Harmonic Drive AG

YukonDrive User Manual

CANopen

EtherCAT Technology Group

Harmonic Drive AG

1003376 09/2011

Table of contents

1 General Introduction....4
1.1 Measures for your safety....4
1.2 Introduction to CANopen....5
1.3 Introduction to EtherCAT....6
1.4 System requirements....6
1.5 Further documentation....6
2 Mounting and Connection of CANopen....7
2.1 Setting the address....7
2.2 Meanings of LEDs....9
2.3 Installation....10
2.4 Transmission speeds....12
2.5 Display of operating states via 7-segment display....13
2.6 Hardware enable....13
3 Mounting and Connection of EtherCAT....14
3.1 Installation and cabling....14
3.2 Pin assignment of the RJ-45 socket....15
3.3 Meanings of LEDs....16
3.4 Display of operating statuses via 7-segment display....17
3.5 Hardware enable....18
4 Commissioning and Configuration of CANopen....19
4.1 General commissioning of CANopen/EtherCAT....19
4.1.1 Commissioning....19
4.1.2 Commissioning sequence....19
4.1.3 Commissioning via DriveManager....20
4.1.4 Operation mode selection (modes of operation)....20
4.1.5 Functionality of operation modes....21
4.1.6 Setting the timing parameters....21
4.2 CAN-specific configuration....21
4.2.1 Setting the software address and baud rate....22
4.2.2 Commissioning instructions....21
4.2.3 Testing the higher-order controller....22
4.2.4 Data handling....23
4.2.5 Control functions....23
4.3 Commissioning and Configuration of EtherCAT....23
5 Setting the Device Parameters for CANopen....23
5.1 Implemented CiA-301 functionality....23
5.1.1 Communication objects....23
5.1.2 Object directory of CiA-301....24
5.2 Parameter channel (Service Data Objects)....26
5.2.1 Data types....26
5.2.2 Representation of data types in the control protocol....27
5.2.3 Access to device parameters....27
5.3 Examples of SDO handling....28
5.3.1 Parameter set download....32
5.4 PDO transmission types....33
5.5 Event-controlled TxPDO transmission....33
5.6 PDO mapping....34
5.6.1 Mapping – general....34
5.6.2 Mapping notes....34
5.7 Heartbeat function....35
5.8 Monitoring of telegram failure....36
6 Setting the Device Parameters for EtherCAT....36
6.1 Supported EtherCAT functionality....36
6.2 Configuration for operation in a controller....40

7 Implemented CiA-402 functionality ... 40
7.1 Device control and state machine ... 40
7.1.1 General information ... 40
7.1.2 State machine ... 41
7.1.3 Device states ... 42
7.2 Option codes ... 44
7.3 Device control objects ... 45
7.4 Units and scalings, factor group ... 46
7.5 I/O map ... 48
7.5.1 Object 60FDh – digital inputs ... 49
7.5.2 Object 2079h – MPRO_INPUT_STATE ... 49
7.5.3 Object 208Fh – MRPO_OUTPUT_STATE ... 50
7.5.4 Setting digital outputs via field bus ... 50
7.5.5 Object 60FE, digital outputs: ... 51
8 Operation modes CiA-402 ... 51
8.1 CiA-402 compatible operation modes ... 51
8.1.1 Configuring YukonDrive for activation via CiA-402 ... 51
8.1.2 Control word CiA-402 ... 52
8.1.3 Status word CiA-402 ... 54
8.2 Operation modes with profile generation in drive ... 56
8.2.1 Profile velocity mode ... 56
8.2.2 Homing mode ... 58
8.2.3 Profile position mode ... 60
8.2.4 Velocity mode (V/F mode) ... 63
8.3 Cyclical operation modes, profile generation in the controller ... 64
8.3.1 Interpolated position mode ... 64
8.3.2 Cyclic Synchronous Position mode (EtherCAT only) ... 65
8.3.3 Cyclic Synchronous Velocity mode (EtherCAT only) ... 66
8.3.4 Cyclic Synchronous Torque mode (EtherCAT only) ... 67
8.3.5 External pre-control of speed/torque ... 68
9 Emergency objects ... 69
9.1 Error acknowledgement, general ... 69
9.2 Error acknowledgement via bus system ... 69
10 Technology functions ... 70
10.1 Touch probe ... 70
10.1.1 Description of manufacturer-specific implementation ... 70
10.1.2 Control-led homing ... 73
10.2 Indexing table function ... 73
11 EDS file, object directory, parameter list ... 73
11.1 EDS file, object directory ... 73
13 Appendix: Glossary ... 74

1 General Introduction

1.1 Measures for your safety

The YukonDrive drive controllers are quick and safe to handle. For your own safety and for the safe functioning of your device, please be sure to observe the following points:

- Follow the safety instructions!

Electric drives are dangerous:

- Electrical voltages > 230 V/460 V:
  Dangerously high voltages may still be present 10 minutes after the power is cut, so always make sure the system is no longer live.
- Rotating parts.
- Hot surfaces.

Your qualification:

- In order to prevent personal injury and damage to property, only qualified electrical engineers may work on the device.
- Knowledge of national accident prevention regulations (e.g. VBG4 in Germany).
- Knowledge of layout and interconnection with the CAN bus field bus.

During installation observe the following instructions:

- Always comply with the connection conditions and technical specifications.
- Electrical installation standards, e.g. for cable cross-section, shielding etc.
- Do not touch electronic components and contacts (electrostatic discharge may destroy components).

1.2 Introduction to CANopen

CANopen is an interconnection concept based on the CAN (Controller Area Network) serial bus system. CAN has many specific advantages, in particular multi-master capability, real-time capability, resistant response to electromagnetic interference, a high level of availability and the low cost of controller chips. These advantages have resulted in CAN being introduced into widespread use in automation too.

Simplified cross-manufacturer communication

The integration of any number of devices in a manufacturer-specific network involves substantial expense. CANopen was developed to solve this problem. In CANopen the use of CAN identifiers (message addresses), the time response on the bus, the network management (e.g. system start and user monitoring) and the coding of the data contents is specified in a uniform way. CANopen makes it possible for devices from different manufacturers to communicate in a network at minimal cost. CANopen uses a subset of the communication services offered by CAL to define an open interface. The selected CAL services are summarised in a kind of "user guide". This guide is called the CANopen Communication Profile.

CANopen functionality of YukonDrive

The CANopen Communication Profile is documented in CiA-301 and regulates the way communication is executed. It distinguishes between process data objects (PDOs) and service data objects (SDOs).
The Communication Profile additionally defines a simplified network management system.

The device profile for CiA-402 (Rev. 2.0) variable-speed drives was compiled on the basis of the CiA-301 (Rev. 4.01) communication services. It describes the operation modes and device parameters supported.

The following sections will provide you with an overview of the CANopen functionality integrated in YukonDrive, followed by the information necessary for commissioning.

1.3 Introduction to EtherCAT

As far as real-time Ethernet systems are concerned, EtherCAT has become well established in the area of automation. The decisive factor here is not only the IEEE802.3 / 100BaseTX Ethernet physics known in the home office area, but also the excellent value for money with regard to implementation in the master and slave modules.

Interconnection can be executed as required in a star, ring or line structure using standard patch or crossover cables and is therefore easily adapted to the machine infrastructure.

To reduce the amount of training required, familiar communication and device profiles were used as of the application layer. In this way, users familiar with CANopen profiles such as CiA-301 or CiA-402 can change over to this new field bus technology with minimal training.

In YukonDrive we have combined all our past experience in the CANopen area with this new field bus technology and achieved maximum compatibility and functionality.

1.4 System requirements

It is assumed you have a standard CANopen setup program and a CANopen interface driver.

For the precise protocol definitions, please refer to the CAL specification.

With the aid of these objects it is possible to configure the actual CANopen communication very flexibly and adapt it to the specific needs of the user.

1.5 Further documentation

- Operation manual, for commissioning of the drive unit
- Application manual, for additional parameter setting to adapt to the application. The application manual can be downloaded as a PDF file from our website at http://www.harmonicdrive.de. Follow the Download link.
- CiA-301 (Rev. 4.0): Application Layer and Communication Profile
- CiA-402 (Rev. 2.0): Device Profile Drives and Motion Control
- EtherCAT Communication Specification Version 1.0 2004
- EtherCAT Indicator Specification Proposal V0.91 2005
- IEC61158-2-12 to IEC61158-6-12

2 Mounting and Connection of CANopen

ATTENTION: Do NOT insert or remove the CANopen connector during operation.

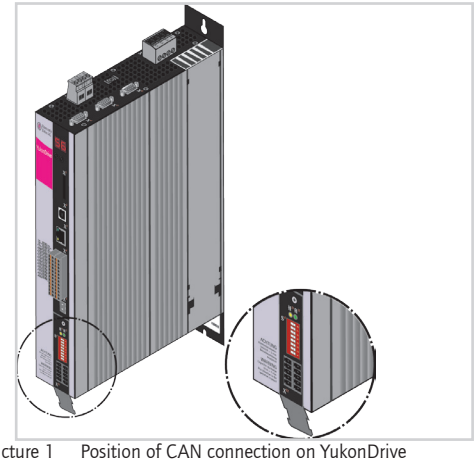
2.1 Setting the address

Table 1 Address setting

Schritt	Aktion	Anmerkung
	Find out which address is assigned to the device you are installing.	Ask your project engineer.
	Select the mode of addressing: • by bus address parameter • by DIP switch (S4) • by bus address parameter and DIP switch (S4)	See below
Address setting finished; for further procedure see Installation		

Three possible methods of address allocation

1. Only using bus address parameter P 2005-COM_CAN_Adr: You will find parameter P 2005-COM_CAN_Adr (factory setting 1) in the “field bus” subject area under CANopen.
2. Only using DIP switch S4
3. Combination of bus address parameter and DIP switch S4 CAN address = hardware address (S4) + parameter P 2005-COM_CAN_Adr. This option is advantageous if, for example, you intend always to use the same parameter set with up to 15 drives, but the lowest address is 30. Parameter P 2005-COM_CAN_Adr is then set to 30. The device address is then defined using the coding switch, which ranges from 0-15.



Picture 1 Position of CAN connection on YukonDrive

address setting using DIP switch

An address between 0 and 63 can be selected decimally using DIP switch S4 on the position controller.

The DIP switch is assigned as follows: Positions are reserved for the address setting, position 1-6/7 for the activation/deactivation of the 120 Ohm bus termination in the device.

Function/assignment:

DIP switch 1 → significance $2^0 = 1$

DIP switch 2 → significance $2^1 = 2$

DIP switch 3 → significance $2^2 = 4$

...

DIP switch 6 → significance $2^5 = 32$

DIP switch 7 = bus termination ON/OFF

Picture 2 Device with CANopen Option

Example of use of the DIP switches:

Setting address "3" using the DIP switches:

- → Set switch 1 and switch 2 to ON
- → $2^0 + 2^1 = 3$
- → Resulting device address = 3
- → (If the software address = 0 is set)

IMPORTANT: Switch 7 = bus termination!

Note: Changes to the CAN address are applied on a
- Reset node command
- Restart (device power-up).

Note: The active bus address can be found in the boot-up message.

2.2 Meanings of LEDs

The CAN option of YukonDrive has two diagnostic LEDs (H14, H15).

Picture 3 Device with CANopen Option

The LEDs have the following function:

Table 2 Meanings of LEDs

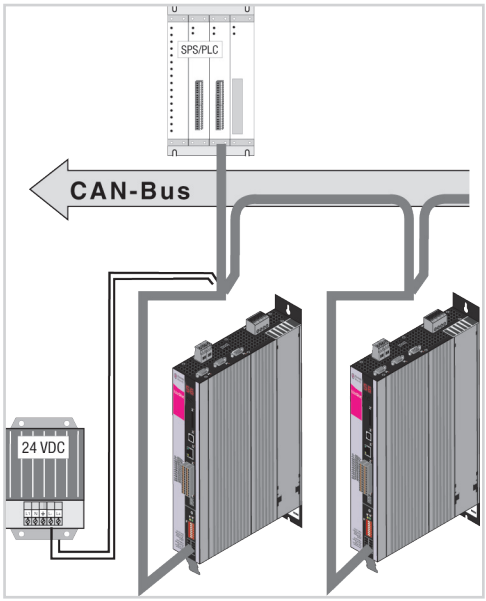
LED	Function	Meaning
H14 (yellow LED)	CANopen network status	The LED displays the current network status. • NMT STOPPED → flashing with 800 ms cycle • NMT PRE-OPERATIONAL → flashing with 1600 ms cycle • NMT OPERATIONAL → permanently lit.
H15 (green LED)	Voltage supply CAN option	Permanently lit if the 24V supply is powering the CAN option via the CAN bus.

2.3 Installation

Table 3 Electrical installation

Step	Action	Note
1	Make sure the hardware enable is wired on YukonDrive (X4).	• See Operation Manual
2	Wire the CAN connection using connector X32 • Connection of CAN signal cables • Connection of interface power supply • Activation of the internal bus terminating resistor on the final drive controller	See Specification of CAN bus connection table and Assignment of connection X19 table
3	Switch on the drive device	
Electrical installation is finished; for how to proceed further, refer to section 4 "Commissioning and configuration".		

The CANopen interface is integrated in YukonDrive. The connection is made via connector X32. The interface is isolated from the drive controller electronics. The supply to the isolated secondary side is provided by the customer via connector X32.

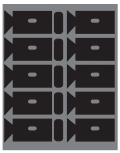


Picture 4 System connection

Table 4 Specification of CAN bus connection

Connection	Spring-type terminal
Wave terminating resistor - Bus termination -	• 120 W (internal) • Activation of the bus termination in the device via switch 8 on the CAN option
Max. input frequency	1 MHz
Ext. voltage supply	+24V +25%, 50 mA (isolated from drive controller)
Voltage ripple	Max. 3 Vss
Power consumption	Max. 50 mA per user
Cable type	4-wire, surge impedance 120 Ω

Table 5 Assignment of connection X32

Terminal X32	PIN	PIN	Function	Description
	10	5	CAN_+24V	External 24V supply
	9	4	CAN_H	CAN High
	8	3	CAN_SHLD	CAN Shield (optional)
	7	2	CAN_L	CAN Low
	6	1	CAN_GND	CAN Ground (0V)

NOTE: Both connectors on terminal X32 are connected to each other in the device.

NOTE: The external 24V supply for the option board is essential. It is not powered by the device.

2.4 Transmission speeds

The CAN bus can be operated at the following baud rates:

Table 6 Transmission speeds

Transmission speed	Mximum line length over the entire network 1)	
1000 kBaud	25 m	• Factory setting
500 KBaud	100 m	
250 kBaud 2)	250 m	
125 kBaud 2)	500 m	
50 kBaud 3)	1000 m	
20 kBaud 3)	2500 m	

1) Rounded bus length estimation (worst case) on basis 5 ns/m propagation delay and a total effective device internal in-out delay as follows:
1M–800 kbit/s: 210 ns
500–250 kbit/s: 300 ns (includes 2 * 40 ns for optocouplers)
125 kbit/s: 450 ns (includes 2 * 100 ns for optocouplers)
50–10 kbit/s: Effective delay = delay recessive to dominant plus dominant to recessive divided by two.

2) For a bus length greater than about 200 m, the use of optocouplers is recommended. If optocouplers are placed between the CAN Controller and the transceiver this affects the maximum bus length depending upon the propagation delay of the optocouplers, i.e. -4m per 10 ns propagation delay of employed optocoupler type.

3) For a bus length greater than about 1 km, bridge or repeater devices may be needed.

When selecting the transmission rate it should, however, be ensured that the line length does not exceed the permissible line length for the transmission rate in question.

2.5 Display of operating states via 7-segment display

Table 7

D1	D2	Meaning	Parameter
System statuses			
8.	*8.*	Device in reset status	
	0.	Auto-initialisation on device startup	(Start)
S.)*	*1.*	1) Not ready to switch on (no DC-link voltage)	(NotReadyToSwitchOn)
S.	*2.*	1) Starting lockout (DC-link is OK, power stage not ready)	(SwitchOnDisabled)
	3.	Ready to switch on (power stage is ready)	(ReadyToSwitchOn)
	4.	On (power is connected to the device) [2]	(SwitchedOn)
	5.	Drive ready (current applied to drive and drive ready for input of setpoint) [2]	(OperationEnable)
	6.	Quick stop [2]	(QuickStopActive)
	7.	Fault response active [2]	(FaultReactionActive)
E	*R*	Fault (see below)	(Fault)
The following appear alternately in the event of error			
E	*R.*	Display for errors or non-acknowledgeable errors	
X	*X*	Error number (decimal)	
Y	*Y*	Error localisation (decimal)	

1) S. flashes if the STO (Safe Torque Off) function is active; the display is not lit if the function is not active.
*) This is not a „safe display" under the terms of EN 61800-5-2.

2) The point flashes if the power stage is active.

Example of the flash sequence:

→ ER > 02 > 05 * ER > 02 > 05 ...

Table 8

	Error:	ER = "Fault"
	Error name:	02 = "Error in the parameter list"
	Description of error:	05 = "Function for checking current parameter list"

2.6 Hardware enable

YukonDrive has a control input for ENPO hardware enable on the control terminal. This input must be configured to operate the power stage at 24V.

The device also provides the function "STO (Safe Torque Off)" (see Operation Manual or Application Manual YukonDrive), category 3, control terminal ISDSH. For these devices the relevant function logic must be implemented by way of the higher-order controller as per the Application Manual.

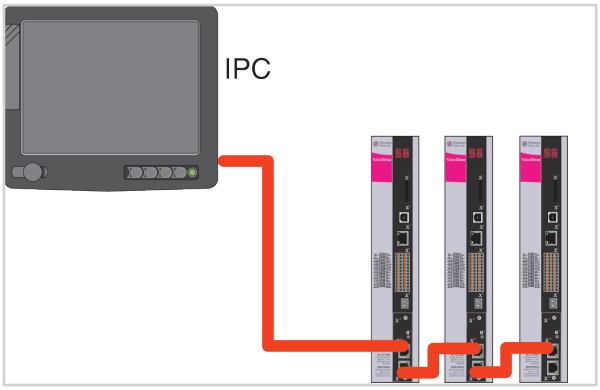
NOTE: When the inputs ENPO and ISDSH are not configured, the device stays in status 1 = "Not Ready to Switch On" or 2 = "Switch On Disabled".
Only after correct configuration can the status be exited by a „Shutdown" command via bus.

3 Mounting and Connection of EtherCAT

3.1 Installation and cabling

Setup of the EtherCAT network

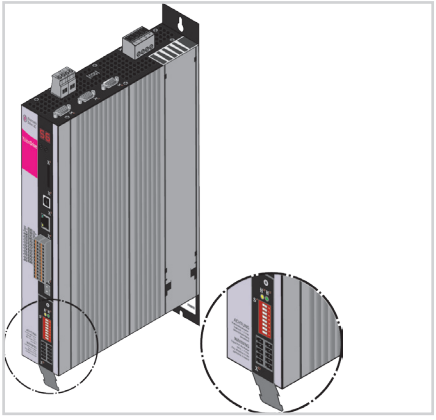
In an EtherCAT network there is always one EtherCAT master (e.g. an industrial PC) and a variable number of slaves (e.g. servo controller, bus terminals etc). Each EtherCAT slave has two Ethernet ports. Slave to slave cabling is thus possible. All EtherCAT users are generally connected in a line with the master at the beginning of the circuit. On the last slave in the line the second Ethernet port remains open. Ethernet-Port offen.



Picture 5 Anschluss EtherCAT

IN and OUT socket (RJ-45 input/output)

Each EtherCAT slave has two RJ-45 sockets. The upper port (X15) is the input (IN) and the lower port (X16) is the output (OUT) of the slave. The incoming cable (from the direction of the master) is connected using the IN port, and the outgoing cable is connected to the next slave using the OUT port. The OUT port remains blank for the last slave in the series. An open output on a slave leads internally to a logical short circuit of the transmit (Tx) and receive (Rx) cables. For this reason every EtherCAT network can be regarded as a logical ring in terms of its topology.



Picture 6 EtherCAT option

Upper RJ-45 port = input

Lower RJ-45 port = output

IMPORTANT: Errors in cabling (incorrect connection of input and output) can lead to faulty addressing by the master.

Connecting cables

Ethernet patch cables or crossover cables are suitable connection cables as per the CAT5e specification. Cables lengths of 0.3 m to a max. 100 m are permissible.

IMPORTANT: Never use EtherCAT and standard Ethernet together in one physical network.
This can lead to impairments including communication outages!
To avoid confusion, always use different colours for EtherCAT and Ethernet cables.

3.2 Pin assignment of the RJ-45 socket

The two LEDs on the RJ-45 socket mean the following:

Table 9 Meaning of LEDs without additional status/error LED

PIN	Farbe	Kabel-Aderpaar	Funktion
1	white/orange	2	TxData +
2	orange	2	TxData -
3	white/green	3	RecvData +
4	blue	1	Unused
5	white/blue	1	Unused
6	green	3	RecvData -
7	white/brown	4	Unused
8	brown	4	Unused

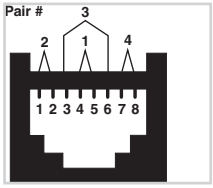
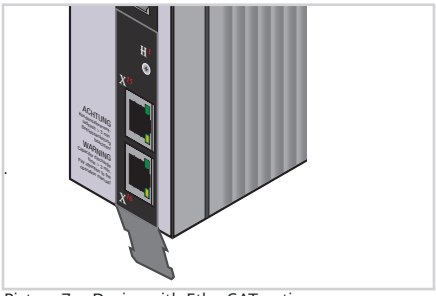


Bild 1.1 RJ-45 Buchse

NOTE: Ethernet cables are available in various lengths in the IT specialist trade. Use CAT5e cable or better.

3.3 Meanings of LEDs

There are 2 LEDs on each RJ-45 socket.



Picture 7 Device with EtherCAT option

The two LEDs on the RJ-45 socket mean the following:

Table 10 Meaning of LEDs without additional status/error LED

LED	Function	Meaning
Upper LED	Link / Activity	Off = no link → No connection with another user
		On = link → Connection with another user exists, no data exchange
		Flashing = activity → Data exchange active
Lower LED	RUN (only active on the lower port if another user is connected here)	Off = initialisation → Device is in initialisation state
		Flashing = pre-operational → Device is in „pre-operational" state
		Single flash = safe-operational → Device is in „safe-operational" state
		On = operational → Device ready to start

Depending on the device's hardware status, an additional status/error LED may be present in addition to the two LEDs on the two RJ-45 sockets. In this case, the meaning of the LEDs is as shown in the table below.

Table 11 Meaning of LEDs with additional status/error LED

LED	Function	Meaning
Upper LED	Link / Activity	Off = no link → No connection with another user
		On = link → Connection with another user exists, no data exchangeh
Lower LED	Link (PHY)	On = link Off = no link

LED	Function	Meaning
Status LED (RUN/ error)	Status/error	Red = error Off = No Error Blinking = Invalid Configuration Single Flash = Local Error Double Flash = Watchdog timeout Green = RUN Off = initialisation → Device is in initialisation state Flashing = pre-operational → Device is in "pre-operational" state Single flash = safe-operational → Device is in "safe-operational" state On = operational → Device ready to start

3.4 Display of operating statuses via 7-segment display

Table 12

D1	D2	Meaning	Parameter
System statuses			
8.	*8.*	Device in reset status	
	0.	Auto-initialisation on device startup	(Start)
S.)*	*1.*	1) Not ready to switch on (no DC-link voltage)	(NotReadyToSwitchOn)
S.)*	*2.*	1) Starting lockout (DC-link is OK, power stage not ready)	(SwitchOnDisabled)
	3.	Ready to switch on (power stage is ready)	(ReadyToSwitchOn)

D1	D2	Meaning	Parameter
	4.	On (power is connected to the device)[2)]	(SwitchedOn)
	5.	Drive ready (current applied to drive and drive ready for input of setpoint) [2)]	(OperationEnable)
	6.	Quick stop [2)]	(QuickStopActive)
	7.	Fault response active [2)]	(FaultReactionActive)
E	*R*	Fault (see below)	(Fault)
The following appear alternately in the event of error			
E	*R.*	Display for errors or non-acknowledgeable errors	
X	*Y*	Error number (decimal)	
X	*Y*	Error localisation (decimal)	

1) S. flashes if the STO (Safe Torque Off) function is active; the display is not lit if the function is not active.
*) This is not a „safe display" under the terms of EN 61800-5-2.

2) The point flashes if the power stage is active.

Example of the flash sequence::

→ ER > 02 > 05 * ER > 02 > 05 ...

Table 13

	Error:	ER = "Fault"
	Error name:	02 = "Error in the parameter list"
	Description of error:	5 = "Function for checking current parameter list"

3.5 Hardware enable

YukonDrive has a control input for ENPO hardware enable on the control terminal.
This input must be configured to operate the power stage at 24V.

The device also provides the function "STO (Safe Torque Off)" (see Operation Manual or Application Manual YukonDrive), category 3, control terminal ISDSH. For these devices the relevant function logic must be implemented by way of the higher-order controller as per the Application Manual.

Note: When the inputs ENPO and ISDSH are not configured, the device stays in status 1 = "Not Ready to Switch On" or 2 = "Switch On Disabled".
Only after correct configuration can the status be exited by a „Shutdown command" via bus.

4 Commissioning and Configuration of CANopen

4.1 General commissioning of CANopen/EtherCAT

4.1.1 Commissioning

The DriveManager user interface is used for general commissioning of the drive system. The DriveManager includes tools to identify motor data, provides access to a motor database for servo motors and enables general device configuration.

First commissioning is a separate subject regarding operation via the user interface and is described in detail in the device's application manual.

4.1.2 Commissioning sequence

Preconditions:

- The drive device is wired as specified in the operation manual and first commissioning is completed. (To test CAN communication, it is sufficient to connect the voltage supply of the CAN option and the control voltage).
- If current is to be applied to the motor, the hardware enable (ENPO) and the "STO (Safe Torque Off)" must also be correctly configured.

NOTE: For more detailed information on optimisation of the software functions and control circuits, refer to the device application manual.

Table 14

Step	Action	Note
1	Check the wiring. Make sure the ENPO hardware enable (X4) is not connected.	
2	Switch on the mains power and the 24V supply to the CAN interface.	
3	Configure the drive device using the application manual.	(Inputs/outputs, software functions etc.)
4	Test the control quality and optimise the controller settings as necessary using the operation manual.	
5	Set the parameters for the CAN communication. The baud rate and the device address are required. The address can be selected using software and hardware. The mapping must also be completed and the active operation mode selected as per CiA-301/402.	Software and hardware address are added...
6	Test the drive on the higher-order controller – see section 3.4.	
7	Finally, save the setting.	Save device setting → Non-volatile in device

NOTE: For more information on the subject of "Units and scalings", please refer to section 7.4.

.

4.1.3 Commissioning via DriveManager

Procedure for commissioning with the aid of the application manual

Table 15

1.	First commissioning based on operation manual	A precondition for this is first commissioning with the aid of the operation manual. The user manual only covers adjustment of the software functions
2.	Commissioning as per application manual	Setting the drive controller parameters using the application manual. This includes, for example, the configuration of technology functions.
3.	Commissioning based on CANopen user manual	Configuration of field bus-specific settings (e.g. baud rate) using this document.
4.	Checking the set application solution	To preserve the safety of personnel and machinery, the application solution should only be checked at low speed. Make sure the direction of rotation is correct. In case of emergency the controller power stage can be disabled, and the drive stopped, by removing the ENPO signal.
5.	Completing commissioning	When you have successfully completed commissioning, save your settings (using DriveManager) and store the data set in the device

4.1.4 Operation mode selection (modes of operation)

There are different control modes for operation of the devices via CANopen. The active operation mode is always selected via CiA-402 object 6060h (Modes of Operation).

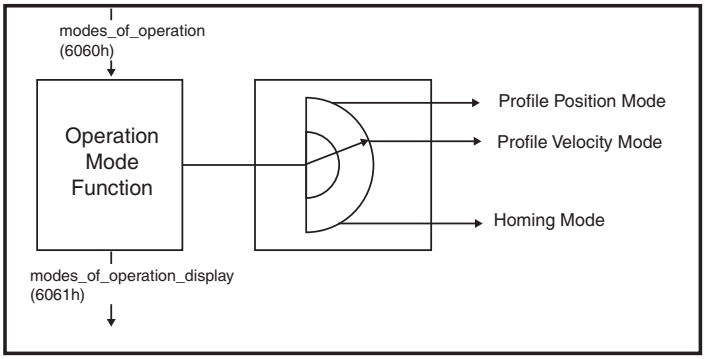
- YukonDrive supports the operation modes as per the CiA-402:
- Profile Position mode
- Profile Velocity mode
- Homing mode
- Interpolated Position mode
- Cyclic Synchronous Position mode (EtherCAT only)
- Cyclic Synchronous Velocity mode (EtherCAT only)
- Cyclic Synchronous Torque mode (EtherCAT only)

In the course of first commissioning the user implements the drive settings using motor data, control settings, I/O configuration etc.

A relevant control mode is also directly connected with the respective operation mode. By switching modes of operation via CANopen/EtherCAT, it is possible to switch directly between position control, speed control and torque control.

The drive is thus in speed control for Profile Velocity mode and in position control for Profile Position mode.

4.1.5 Functionality of operation modes



Picture 8 Changing functionality of operation modes in the display

Users can switch between the various operation modes, as long as these are supported by the device.
The status word contains bits, the meaning of which depends on the operation mode. For monitoring, it is necessary for the bits to change their meaning when switching operation modes; see also Chapter 6.

4.1.6 Setting the timing parameter

To ensure correct communication with the controller, three timing parameters have to be set on YukonDrive. As a rule, these should all be set to the same value. It should be borne in mind that different units have to be used when setting the three parameters (see table).

Tabelle 16 Setting the timing parameters

Para – ID	Name / Object	Unit
306	Internal interpolator cycle time	ms
2015	Communication cycle period/0x1006	µs
2266 Index 0 Index 1	0x60C2 Interpolation time base Interpolation time exponent	 s -

For EtherCAT, parameter P 2266 MPRO_402_IntTimePeriod (object 0x60C2) must be set to the cycle time for the setpoints (or the telegrams).

4.2 CAN-specific configuration

4.2.1 Setting the software address and baud rate

The software address and baud rate can be set using the following device parameters via DriveManager:

Tabelle 17 Parameters on the Bus Systems function screen

Parameter	Function	Description
2005-COM_CAN_Adr	CANopen address	Address assignment via parameter. For more information on setting the address, see section 2.1
2006-COM_CAN_Baudrate	Baudrate	Permissible baud rates – see section 2.3

NOTE: YukonDrive has a default baud rate of 1 Mbit.
The actual address is calculated by adding the software and hardware address and is displayed using parameter P 2058 COM_CAN_Adr_Act.

Any change to the baud rate in parameter P 2006 COM_CAN_Baudrate only takes effect once YukonDrive has been restarted. The current baud rate is displayed using parameter P 2059 COM_CAN_Baudrate_act.

4.2.2 Commissioning instructions

A drive device may not respond to a telegram for a variety of reasons:

- There is no reply if the scope of telegram (baud rate, data length) on the master computer is not correct.
- There is no reply if a drive device is addressed with the wrong bus address.
- There is no reply if the serial connection between the master computer and the drive device is not correctly set up.
- There is no reply if the 24V supply to the CAN connection is missing or the cabling is faulty.
- There is no valid reply if several devices with the same device address are connected to the bus.
- There is no reply if the device has certain network statuses. The current network status can be checked using parameter P 2060 COM_CAN_NMT_State.

Tabelle 18 Parameter 2060

Parameter 2060	Description
0	Bootup
1	Init
4	Stopped / Safe OP
5	Operational
127	Pre-Operational

4.2.3 Testing the higher-order controller

To activate changed settings the device must be switched off and back on again. When the power is connected, after an initialisation period of a few seconds the device must transmit a one-off boot-up message (ID 700h + node ID = 701h for device address 1).

If this happens, the communication is OK.

NOTE: When transferring data to the device via SDO telegrams the number of data bytes transferred should be taken into account. For this the correct length information must be transferred in the control byte. Alternatively, however, an SDO transfer without specification of the data length is also possible. The correct operation of the control byte in the SDO telegram should also be observed.

4.2.4 Data handling

Saving the settings

All configuration data can be backed up by the DriveManager.

NOTE: Please note, however, that some objects are RAM variables, which must be correctly operated and initialised by the controller. This includes, for example, object 6060h: Modes of Operation.

Restoring factory defaults

There are two possible ways of restoring the devices' default factory settings:

Via field bus

- Write value 1 to subindex 3 of object 200BH-PARA_SetCmd. The factory settings are then applied to the whole device.

NOTE: Please note that this also affects the settings for the baud rate/device address. The changes take effect after a "Reset node" command or device restart.

Via DriveManager

- First select the relevant YukonDrive in the DriveManager tree structure. The right mouse button opens a context menu from which you can select the "Reset Device Setting" entry.

NOTE: In both cases it takes approx. 10 seconds for the device to signal that it is ready for operation again. During this time the device performs a self-test and changes all its settings to the factory setting. However, this setting is only retained if the data is backed up in the device. Data backup is initiated via the DriveManager user interface or by writing to object 200BH-PARA_SetCmd Subindex 1 = 1 via the bus system. The save operation can also be executed using object 1010 hex.

ATTENTION: Data backup takes a few hundred ms. During that time the device must not be switched off, otherwise the settings will be lost.

Object 200BH-PARA_SetCmd Subindex 1 is automatically set to 0 by the device after the save operation. This process can be used for timeout monitoring of the function.

4.2.5 Control functions

Control functions can be optimally adapted to the relevant application. Consequently, several control formats are offered. The appropriate formats can be selected by the master during the setup phase via the bus, or by adjusting the relevant device parameters.

The drive devices' state machine has a cycle time of 1 ms.

All control commands and setpoints are processed within that cycle time by the drive device.

NOTE: Control PDOs are processed in a minimum cycle time of 1 ms. If protocols arrive at the device faster, the telegram that arrived most recently overwrites the previous one. An error message is not generated if telegrams are overwritten as a result of insufficient cycle time.

4.3 Commissioning and Configuration of EtherCAT

Commissioning via EtherCAT is possible using the XML file supplied on your controller. All further commissioning and configuration steps depend on the controller used. For notes on this, please refer to the documentation provided by your controller manufacturer.

5 Setting the Device Parameters for CANopen

5.1 Implemented CiA-301 functionality

5.1.1 Communication objects

- Boot-up to CiA-301 V4.01 (guarding boot-up via identifier 700h)
- Four variably mappable TxPDOs (transmission type 1 to 240, 254 and 255 dec possible)
- Four variably mappable RxPDOs (transmission type 1 to 240, 254 and 255 dec possible)
- One SDO server – pay attention to definition of time conditions (typical processing time in device approx. 5 ms, depending on capacity utilisation)
- One emergency object error code to CiA-402 plus manufacturer-specific error location and number, operating hours of the device
- One Sync object
- NMT state machine to CiA-301
- Node guarding and heartbeat (see below)
- Processing cycle:
  PDO protocols can be processed in a minimum cycle time of 1 ms. If protocols arrive faster, the previous protocols are overwritten.
- SDO protocols and NMT services are processed acyclically. Typical processing times lie between 1 and 5 ms.
- Initialisation values of the COB IDs based on Predefined Connection Set
- Access to device parameters 2000h–5FFFh (expedited/non-expedited)

5.1.2 Object directory of CiA-301

For a full overview of the supported CAN objects of YukonDrive, please refer to the EDS file.

Here you can refer both to the CANopen objects of CiA-301 and CiA-402, and to the manufacturer-specific objects of the device.

The following list shows an extract of the object directories with important CiA-301 objects. For these objects the transmission types or mapping, for example, are explained below.

Table 19 Object directory

Objekt Nr.	Objekt Name	Objekt Code	Typ	Attr.
0x1000	Device_Type	VAR	Unsigned32	ro
0x1001	Error_Register	VAR	Unsigned8	ro
0x1003	Pre-Defined_Error_Field One subentry	ARRAY	Unsigned32	ro
0x1005	COB-ID_SYNC	VAR	Unsigned32	rw
0x1006	Communication_Cycle_Period	VAR	Unsigned32	rw
0x1007	Synchronous_Window_Length	VAR	Unsigned32	rw
0x1008	Manufacturer device name	String		
0x1009	Manufacturer hardware version	String		
0x100A	Manufacturer software version	String		
0x100C	Guard_Time	VAR	Unsigned16	
0x100D	Life_Time_Factor	VAR	Unsigned8	
0x1010	Store parameters	ARRAY	Unsigned32	rw
0x1011	Restore default parameters	ARRAY	Unsigned32	rw
0x1014	COD-ID_EMCY	VAR	Unsigned32	
0x1017	Producer_Heartbeat_Time	VAR	Unsigned16	rw
0x1018	Identity_Object alle 4 Einträge (Seriennummer, ...) unterstützen	RECORD	Identity (23h)	ro
0x1400	1st_Receive_PDO_Parameter	RECORD	PDO CommPar	rw
0x1401	2nd_Receive_PDO_Parameter	RECORD	PDO CommPar	rw
0x1402	3rd_Receive_PDO_Parameter	RECORD	PDO CommPar	rw
0x1403	4th_Receive_PDO_Parameter	RECORD	PDO CommPar	rw

Table 20 Object directory

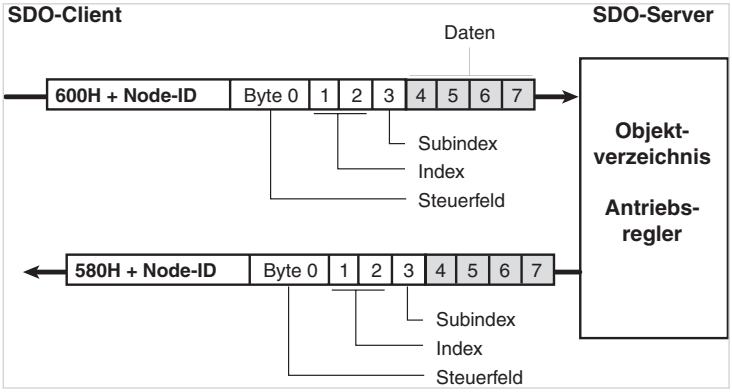
Objekt Nr.	Objekt Name	Objekt Code	Typ	Attr.
0x1600	1st_Receive_PDO_Mapping max 8 objects	RECORD	PDO Mapping (21h)	rw
0x1601	2nd_Receive_PDO_Mapping max 8 objects	RECORD	PDO Mapping	rw
0x1602	3rd_Receive_PDO_Mapping max 8 objects	RECORD	PDO Mapping	rw
0x1603	4th_Receive_PDO_Mapping			
max 8 objects		RECORD	PDO Mapping	rw
0x1800	1st_Transmit_PDO_Parameter	RECORD	PDO CommPar (20h)	rw
0x1801	2nd_Transmit_PDO_Parameter	RECORD	PDO CommPar (20h)	rw
0x1802	3rd_Transmit_PDO_Parameter	RECORD	PDO CommPar	rw
0x1803	4th_Transmit_PDO_Parameter	RECORD	PDO CommPar	rw
0x1A00	1st_Transmit_PDO_Mapping			
max 8 objects		RECORD	PDO Mapping	rw
0x1A01	2nd_Transmit_PDO_Mapping			
max 8 objects		RECORD	PDO Mapping	rw
0x1A02	3rd_Transmit_PDO_Mapping			
max 8 objects		RECORD	PDO Mapping	rw
0x1A03	4th_Transmit_PDO_Mapping			
max 8 objects		RECORD	PDO Mapping	rw

5.2 Parameter channel (Service Data Objects)

The Service Data Object (SDO) permits write and read access to the object directory. This SDO is implemented according to the CAL specification by the Multiplexed Domain CMS object. The protocol is designed for the transfer of data of any length. An SDO server is integrated into the device for SDO transfer. Communication is by way of two reserved identifiers.

Receive SDO: 600 h

Transmit SDO: 580 h



Picture 9 Example of an SDO data transfer in Expedited mode

The CAL specification makes a basic distinction between three protocol services:

- Download protocol (Write)
- Upload protocol (Read)
- Abort protocol (Error)

The upload and download protocols also differentiate between:

- Expedited Multiplexed Domain protocol, for access to objects with a data length of up to 4 bytes (shown above) and
- Multiplexed Domain protocol, for access to objects of any length

The entries in the "Control field" are generated by the CANopen driver. They are only included to fully document the examples cited. The entries are dependent on the transferred data.

The control field is described in the CiA-301 profile.

5.2.1 Data types

NOTE: The DriveManager user interface displays many parameter settings in the form of value substitution texts. Example: Parameter 45 0-MOT_Type = PSM

When writing and reading via the field bus the corresponding numerical values for these value substitution texts must be used. These values are displayed in brackets ( ) when the parameter is opened in DriveManager.

Example:
Parameter 45 0-MOT_Type = PSM (1)

The drive units support the following parameter data formats:

Table 21 Data types

Data type	Value range	Function
USIGN8	0...255	Unsigned
USIGN16	0...65535	
USIGN32	0...4294967295	
INT8	-128...127	Integer, signed
INT8	-32768...32767	
INT32	-2147483648...2147483647	
FLOAT32	see IEEE	32-bit floating point number in IEEE format
STRING		ASCII characters, max. 100 bytes in bus mode incl. zero terminator

5.2.2 Representation of data types in the control protocol

All data types are represented as 32-bit variables in Intel format, and with the correct preceding sign.

Table 22 Assignment of data types in the data field

Daten bytes in the control protocol	3	4	5	6
USIGN8 / INT8* USIGN16 / INT16* USIGN32 / INT32	Low Word LowByte	Low Word High-Byte	High Word Low-Byte	High Word HighByte
FLOAT32	IEEE-Format			
STRING	See examples			
* filled up with the appropriate preceding sign (00H or FFH)				

5.2.3 Access to device parameters

Where can I find the device parameters?

All device parameters are addressed by way of a parameter number.

In addition to the standard objects, the CANopen profile also provides an area for manufacturer-specific entries. This area lies between 2000 h and 5FFF h. If you then want to read or write parameter 455-MOT_FNOM (rated motor frequency) of the device, the object index is generated from 2000 h + parameter number (hex).

In our example: Index = 2000 h + 1C7 H

NOTE: Profile-specific parameters are visible in DriveManager, but only in the 1000H... (CiA-301 objects)/6000H... (CiA-402 objects) writeable/readable range. This means parameters stored both as device parameters (2xxxH range) and as profile parameters (CiA-301/CiA-402) can only be read and written to via their object number (CiA-301/CiA-402 profile).

Example:

The object 1000h Device Type exists both in the CiA-301 profile and also as a device parameter with parameter number 2011. Simultaneous two-way access would therefore be possible via CANopen or EtherCAT. In order to uniquely configure the access, the read/write access for this object is only possible via profile-specific object number 1000h (as per CiA-301).

5.3 Examples of SDO handling

The CANopen objects and the drive controller parameters can be accessed via the Receive SDO (COB IDs: 600 h + node ID).

In a data transfer protocol a maximum of 4 data bytes can be transferred in Expedited mode. This means all device parameters, apart from String parameters, can be written to with a single transfer protocol.

String parameters can be written to using the Multiplexed Domain protocol.

Notes:

Example of read access to string parameters (parameter 3 DV_DeviceAliasName)

Note:
- All numeric values are hexadecimal
- The string „X-axis" is to be transferred
- This text is entered in YukonDrive parameter 3 DV_DeviceAliasName

Table 23

TIME	ID	Direction	DLC	Byte 0	Byte 1	Byte 2
18.992445	Tx	601	8	40	03	20
18.992972	Rx	581	8	41	03	20
35.514341	Tx	601	8	60	00	00
35.514594	Rx	581	8	00	58	2d
36.269620	Tx	601	8	70	00	00
36.270175	Rx	581	8	10	00	00
36.982385	Tx	601	8	60	00	00
36.982664	Rx	581	8	00	00	00
37.686447	Tx	601	8	70	00	00
37.686706	Rx	581	8	10	00	00
38.421344	Tx	601	8	60	00	00
38.421604	Rx	581	8	00	00	00
39.053526	Tx	601	8	70	00	00
39.053787	Rx	581	8	10	00	00
39.749081	Tx	601	8	60	00	00
39.749347	Rx	581	8	00	00	00
40.428981	Tx	601	8	70	00	00
40.429249	Rx	581	8	10	00	00
41.085839	Tx	601	8	60	00	00
41.086198	Rx	581	8	00	00	00
41.740755	Tx	601	8	70	00	00
41.741148	Rx	581	8	10	00	00
42.514034	Tx	601	8	60	00	00
42.514294	Rx	581	8	00	00	00
43.172512	Tx	601	8	70	00	00
43.172787	Rx	581	8	10	00	00
43.908571	Tx	601	8	60	00	00
43.908831	Rx	581	8	00	00	00
44.668466	Tx	601	8	70	00	00
44.668740	Rx	581	8	10	00	00
53.884044	Tx	601	8	60	00	00
53.884414	Rx	581	8	0b	00	00

Transmission of transferred values (ASCII):

At 6 bytes, the string "X-axis" is so short that it can be fully transferred in the first segment.

The following segments (of 100 bytes of the parameter) therefore only include zeroes...

Transmitted bytes (HEX)	58	2d	41	78	69	73
Interpretation (ASCII)	X	-	a	x	i	s

Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Comments
00	00	00	00	00	Read Objekt 2003h (= Parameter 3)
00	64	00	00	00	Reply: 64h --> 100 bytes are to be transferred
00	00	00	00	00	Requirement: segment 1
41	78	69	73	00	Reply: segment 1 - contains „X-Axis"
00	00	00	00	00	Requirement: segment 2
00	00	00	00	00	reply: segment 2
00	00	00	00	00	Requirement: segment 3
00	00	00	00	00	Reply: segment 3
00	00	00	00	00	Requirement: segment 4
00	00	00	00	00	Reply: segment 4
00	00	00	00	00	Requirement: segment 5
00	00	00	00	00	Reply: segment 5
00	00	00	00	00	Requirement: segment 6
00	00	00	00	00	Reply: segment 6
00	00	00	00	00	Requirement: segment 7
00	00	00	00	00	Reply: segment 7
00	00	00	00	00	Requirement: segment 8
00	00	00	00	00	Reply: segment 8
00	00	00	00	00	Requirement: segment 9
00	00	00	00	00	Reply: segment 9
00	00	00	00	00	Requirement: segment 10
00	00	00	00	00	Reply: segment 10
00	00	00	00	00	Requirement: segment 11
00	00	00	00	00	Reply: segment 11
00	00	00	00	00	Requirement: segment 12
00	00	00	00	00	Reply: segment 12
00	00	00	00	00	Requirement: segment 13
00	00	00	00	00	Reply: segment 13
00	00	00	00	00	Requirement: segment 14
00	00	00	00	00	Reply: segment 14
00	00	00	00	00	Requirement: segment 15
00	00	00	00	00	Reply: segment 15 - No further segments

5.3.1 Parameter set download

The following data can be transferred to YukonDrive via the CANopen interface:

- Parameter set
- A parameter data set can be downloaded by SDO transfer or via the DriveManager user interface version 5 or higher. All manufacturer-specific device parameters are also accessible via objects 2000h–5FFFh

If a unified valid data set (i.e. not just individual parameters) needs to be transferred from the CAN master to the device, the following points must be considered:

On every transfer of an individual parameter the drive controller checks whether the parameter matches its existing data set. The check of the new parameter value sometimes refers to existing parameter values. This means it is possible that the drive controller may reject a parameter, even though it originates from a valid parameter data set, because the parameter set is not yet complete in the device.

Since a simple error reset may not eliminate the cause of the error, it may be necessary to reset to the factory defaults.

Remedy:

The parameter data set is transferred to the drive controller without a logic check. At the end of the download, the logic check is reactivated and the drive controller checks the transferred parameters for plausibility. During this check parameter settings that do not functionally match are reported as errors.

Download procedure for a completed parameter data set:

1. Reporting a download without logic check
   To deactivate the logic check and to report the download of a data set, the value 1 is written to parameter 11 subindex 4.
2. Downloading the parameter data to the drive controller
   In this step the individual parameters of the data set are sequentially transferred to the drive. Despite the deactivated logic check, basic checking mechanisms are still active. These monitor, for example, the maintenance of parameter limits and become active if these are infringed. Therefore, if a value range limit is infringed by the download of a parameter, this SDO protocol is directly rejected (Abort message).
3. Completing download and activating plausibility check
   Once all parameter data has been transferred to the drive controller, parameter 11 subindex 4 is reset to the value 0. Then a logic check of the device parameters is carried out. In case of error the user receives an Emergency message.

NOTE: The download of a complete parameter data set is only possible when the system is at a standstill. Make sure the drive controller is not switched on for the duration of the download.

5.4 PDO transmission types

In connection with the PDO transfer, various transmission types are defined in CANopen profile CiA-301. The transmission type and event control can be set separately for all supported RxPDOs and TxPDOs. The drive controller supports the following transmission types:

Acyclic synchronous type no. 0 h

Meaning: RxPDOs are evaluated once a device-specific event has been triggered and the next SYNC object has been received; the TxPDO is then transmitted (from firmware version 2.15-00).

Cyclic synchronous types no. 1–F0 h

Meaning: The difference between this and the acyclic synchronous transmission type is that RxPDOs are only evaluated after receipt of 1–F0 h Sync objects and TxPDOs are only transmitted every 1–F0 h Sync objects.

Asynchronous types no. FE h and FF h

Meaning: RxPDOs are evaluated immediately on receipt; TxPDOs are transmitted by a device-specific event. The Sync object is irrelevant to this mode of transfer. Special feature of type FF h:
For this the event is defined in the associated device profile.

NOTE: The desired transmission types are set by way of the corresponding CANopen objects 1400h for RxPDOs and 1800h for TxPDOs.

5.5 Event-controlled TxPDO transmission

Note: Event control is only active when the relevant "transmission type" is set to asynchronous (FEh or FFh).

Function of event control:

Any bit changes within the TxPDO can serve as an event for the transmission of a TxPDO. This means that only the mapped contents of this TxPDO can be used as an event for transmission of a TxPDO. Accordingly it is not possible to send a TxPDO dependent on the changes in content of another TxPDO.

Example:

The status word 6041h is mapped in TxPDO1. TxPDO2 contains the current actual position. A change in the status word in TxPDO1 can therefore not be used as an event for transmission of the TxPDO2. If this is required, the status word 6041h can also be mapped in TxPDO2 however.

Selecting events:

In YukonDrive every bit (or any change to it) in a TxPDO can be defined as an event. By default all bits (max. 64bit = 8byte) are monitored for changes and are evaluated as events. Individual bits can be displayed using screens, however, and therefore are no longer used for event generation.

Screens enabling the display of individual bits of TxPDOs are defined in field parameter 2007. Each TxPDO has subindexes, and each subindex is responsible for 32 bits of the TxPDO. Its structure is thus as follows:

Parameter 2007 – COM_301_EvMask
„Event mask for asynchronous transmit PDOs"

Table 24 Field parameter 2007

Sub ID	Name	Value	Description	Type
0	EvMsk_TxPdo1L	FFFFFFFFh	Event mask for TxPDO 1 byte 0–3	uint32
1	EvMsk_TxPdo1H	FFFFFFFFh	Event mask for TxPDO 1 byte 4–8	uint32
2	EvMsk_TxPdo2L	FFFFFFFFh	Event mask for TxPDO 2 byte 0–3	uint32
3	EvMsk_TxPdo2H	FFFFFFFFh	Event mask for TxPDO 2 byte 4–8	uint32
4	EvMsk_TxPdo3L	FFFFFFFFh	Event mask for TxPDO 3 byte 0–3	uint32
5	EvMsk_TxPdo3H	FFFFFFFFh	Event mask for TxPDO 3 byte 4–8	uint32
6	EvMsk_TxPdo4L	FFFFFFFFh	Event mask for TxPDO 4 byte 0–3	uint32
7	EvMsk_TxPdo4H	FFFFFFFFh	Event mask for TxPDO 4 byte 4–8	uint32

Example of application of the screens:

To only allow the lower 16 bits of the TxPDO1 as an event, the subindexes of parameter 2007 are described as follows:

Subindex 0 (event mask TxPDO1 bytes 0–3) = 0000FFFFh

Subindex 1 (event mask TxPDO1 bytes 4–7) = 00000000h

NOTE: The cyclic transmission of the TxPDOs is activated by setting a cycle time in ms in the objects 0x1800 (TxPDO1), 0x1801(TxPDO2), 0x1802 (TxPDO3) and 0x1803 (TxPDO4) subindex 5 (event timer).

5.6 PDO mapping

5.6.1 Mapping – general

Variable mapping of parameters is possible on the YukonDrive for all four RxPDOs and TxPDOs. Mapping works as defined in the CANopen communication profile CiA-301.

Most device-specific parameters form part of the manufacturer-specific area (2001h–5FFFh) and can also be mapped in one of the PDOs. For these parameters (objects), refer to the EDS file of the drive controller

5.6.2 Mapping notes

Unlike earlier devices YukonDrive no longer has predefined mapping or mapping selectors. This means that the controller must write the mapping to the drive controller prior to a communication via PDO. Transfer of the data set is also possible.

By default all mapping settings are set to 0, i.e. the PDOs do not contain any mapping. The communication settings (mapping/transmission types etc.) can be saved in the device, however, and are subject to data set handling. This means they do not have to be rewritten each time and can be transferred with the data set.

The following objects are relevant for mapping:

RxPDOs:
1600h RxPDO1 mapping
1601h RxPDO2 mapping
1602h RxPDO3 mapping
1603h RxPDO4 mapping

TxPDOs:
1A00h TxPDO1 mapping
1A01h TxPDO2 mapping
1A02h TxPDO3 mapping
1A03h TxPDO4 mapping

NOTE: A maximum of 8 objects can be mapped per PDO. In a PDO a maximum of 8 bytes can be mapped.

NOTE: Remember that the PDO must always be assigned an even number of bytes! If an uneven number is required, this must be completed with a “dummy byte” for example. Parameter 2055 “COM_301_U8” (object 0x2807) is available for this purpose.

5.7 Heartbeat function

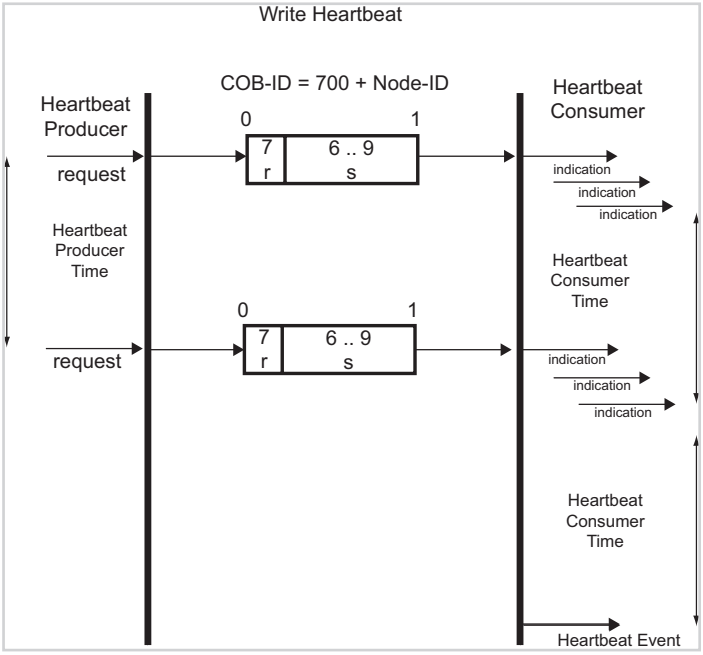
The Heartbeat function according to CiA-301 (V4.01) is supported. YukonDrive can then only be used as a heartbeat producer, i.e. it sends heartbeat telegrams to the controller. To this end object 1017H Producer Heartbeat Time is implemented.

A time value (in ms) is entered as a value for this object. The time value represents the cyclic interval during which the drive controller sends its heartbeat telegrams.

Heartbeat protocol
The Heartbeat protocol defines an ERROR CONTROL SERVICE without using REMOTE FRAMES. A HEARTBEAT PRODUCER sends a cyclic HEARTBEAT MESSAGE. One or more HEARTBEAT CONSUMERS receive this message. The relationship between the PRODUCER and the CONSUMER can be configured by way of the objects described below. The HEARTBEAT CONSUMER monitors receipt of the HEARTBEAT PROTOCOL taking account of the preset HEARTBEAT CONSUMER TIME.

If the HEARTBEAT PROTOCOL is not received within the HEARTBEAT CONSUMER TIME,
a HEARTBEAT event is generated.
The HEARTBEAT PROTOCOL starts directly after entry of the HEARTBEAT PRODUCER TIME. If the device is powered up with a HEARTBEAT PRODUCER TIME setting not equal to 0, the HEARTBEAT PROTOCOL starts with the status transition INITIALISING -> PRE-OPERATIONAL.

In this case the BOOTUP MESSAGE is classed as the first HEARTBEAT MESSAGE.



Picture 10 Heartbeat protocol

r: reserved (always 0)
s: the status of the Heartbeat producer
0: BOOTUP
4: STOPPED
5: OPERATIONAL
127: PRE-OPERATIONAL

NOTE: The NODE GUARDING and HEARTBEAT functions cannot be used in a device simultaneously. If the HEARTBEAT PRODUCER TIME is not equal to 0, the HEARTBEAT PROTOCOL is used.

5.8 Monitoring of telegram failure

YukonDrive can be used to monitor the incoming SYNC telegrams and RxPDOs and to trigger an error message after a configurable number of failed telegrams.

The two parameters shown in the following table are used to configure monitoring:

Table 25 Data Types

Para ID	Name	Description
P 2061	COM_CAN_Timeout_Type	Selection of signal to be monitored: 0: SYNC, 1: RxPDO
P 2062	COM_CAN_Timeout_Value	Timeout time [ms] 0 = monitoring inactive

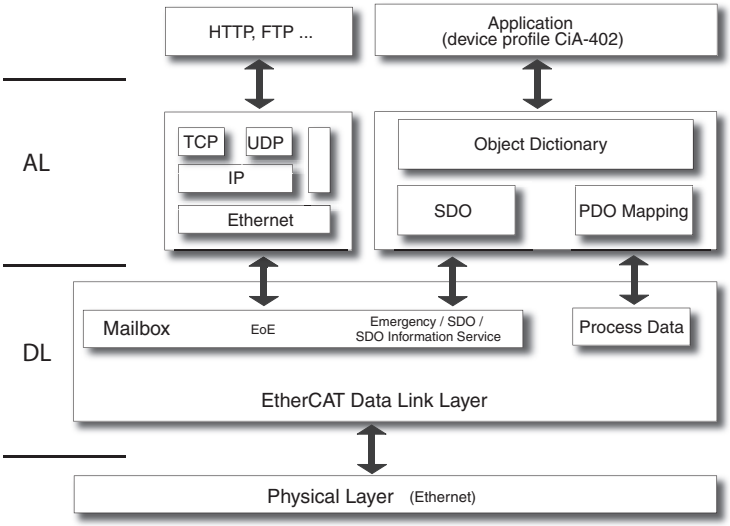
Parameter P 2061 COM_CAN_Timeout_Type can be used to select whether the incoming SYNC signals or the RxPDOs are to be monitored. Parameter P 2062 COM_CAN_Timeout_Value specifies the minimum time in milliseconds that must lapse after the last configured signal before a telegram failure is identified.

Telegram failure monitoring is only active in the NMT status "Operational".

6 Setting the Device Parameters for EtherCAT

6.1 Supported EtherCAT functionality

Below you will find an overview of the EtherCAT functionality implemented in YukonDrive. The diagram below shows the basis for the description that follows. It shows the structure of EtherCAT based on the OSI 7 layer model.



Picture 11 EtherCAT structure

The physical layer of EtherCAT based on IEEE802.3/100 BaseTX Ethernet physics. The EtherCAT Data Link Layer (DL) is then based on this, and is split into mailbox and process data. The following layer is called the Application Layer (AL) and includes the services CoE (CAN over EtherCAT) and EoE (Ethernet over EtherCAT).

All services that are not time-sensitive, i.e. intervention of their execution/contents in process data is not time-sensitive, are grouped together in the mailbox. The mailbox is used as a service data channel and thus also enables access to drive parameters. This is done via the SDO (Service Data Objects) channel. The mailbox service also provides the basis for EoE (Ethernet over EtherCAT) services and error handling (emergency telegrams).

The process data is structured on the basis of CANopen (CiA-301). This means objects are mapped in PDOs (Process Data Objects) that are transferred cyclically. This process data includes, for example, cyclic position, speed or torque reference values and actual values.

The basis for both SDO and PDO accesses to the drive is always the object directory, which is also based on CANopen. For the user this means that these objects can be
accessed both via CANopen and via EtherCAT.

In the case of YukonDrive the CiA-402 device profile is again set up on the application layer. For information on this layer, please refer to the sections “Implemented CiA-402 functionality” and “CiA-402 operation modes”.

An overview of the EtherCAT functionality of YukonDrive is provided below:

Process data

- 4 RxPDOs
- 4 TxPDOs
- Transfer length = max. 8 bytes per PDO
- Variable mapping as per CiA-301 (cf. CANopen)

ATTENTION: The PDO must have an even number of bytes assigned.
If an uneven number is required, this must be completed with a „dummy byte“ for example. The dummy byte is entered as object 0x2807.
Cycle times
Transfer of cyclic position setpoints at max. 8 kHz (125 µs)
Transfer of cyclic speed setpoints at max. 8 kHz (125 µs)
Transfer of cyclic torque setpoints at max. 8 kHz (125 µs)

Mailbox

YukonDrive supports the CAN over EtherCAT (CoE) and Ethernet over EtherCAT (EoE) protocol. The following functions/services are implemented:

CoE

- SDO/Abort
- Initiate SDO Download
- Download SDO Segment
- Initiate SDO Upload
- Upload SDO Segment
- Abort SDO Transfer
- All device parameters are accessible via object ID 2000H + x

Example

The object 1000h Device Type exists both in the CiA-301 profile and also as a device parameter with parameter number 2011. Simultaneous two-way access would therefore be possible via CANopen or EtherCAT. In order to uniquely configure the access, the read/write access for this object is only possible via profile-specific object number 1000h (as per CiA-301).

Emergency

The Emergency service is designed for the transfer of error messages. In contrast to CANopen, emergency messages in EtherCAT are not autonomously sent from the slave but are retrieved by the master.

Functionality in YukonDrive:

- Error codes as per the CiA-402 device profile are supported.
  For the structure/content of the emergency message, please refer to the section
- "Emergency Objects"

SDO Information Service

The SDO Information Service allows the master to read the object directory of the slave. In this way, the master can determine the supported objects of the slave with the required additional information (e.g. data type/access rights etc.). The SDO Information Service therefore represents an alternative to the use of EDS files familiar from CANopen.

Functionality in YukonDrive:

- Access to the object list and description
- Alternative to integrating the EDS file

EoE

Functions such as the tunnelling of standard Ethernet frames in EtherCAT frames generally fall under Ethernet over EtherCAT. This enables protocols, for example TCP/IP, to be transferred via EtherCAT.

Implemented functionality in YukonDrive:

- Initiate EoE request
- Initiate EoE response
- EoE fragment request
- EoE fragment response

Distributed clocks

Synchronisation in EtherCAT is implemented on the basis of distributed clocks. Each slave has its own clock, which is synchronised with the others using a synchronisation pulse. The reference clock with which users are synchronised is accommodated in a slave.

NOTES on YukonDrive:

- All configuration of distributed clocks takes place in the controller.
- Multiples of 125 µs (time basis for control) must always be used as cycle times.

XML file

The XML file is used to integrate an EtherCAT slave into an EtherCAT master (control). It includes the configuration (mapping etc.) for the respective operation modes.

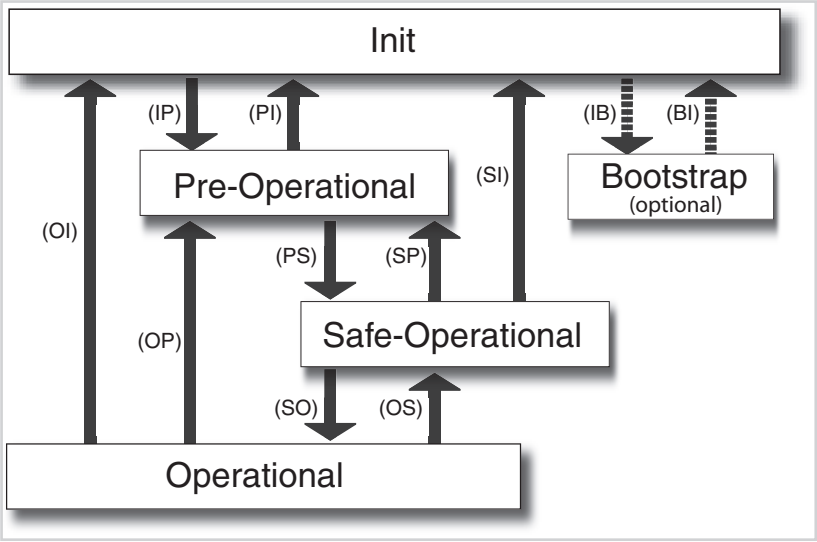
NOTES on YukonDrive:

- The XML file is supplied with the firmware.
- Integration of this file is control-specific.

NMT (Network Management)

The Network Management is essentially based on the network management of CANopen, although Stopped (CANopen) status has been replaced by Safe Operational (EtherCAT) status.

Depending on the range of functions offered by the control software, individual status transitions can be executed automatically or via the PLC.



Picture 12 EtherCAT state machine

Table 26 Status description

Status	Description
Init	Initialisation: the device starts up.
Pre-Operational	The device is ready to be configured. Mailbox communication is possible.
Safe-operational	PDO input data (TxPDO device) can be read. PDO output data (RxPDO device) is ignored.
Operational	Cyclic I/O communication PDO output data (RxPDO device) is processed.

Table 27 Status transitions

Transitions	Operations
IP	Start Mailbox Communication
PI	Stop Mailbox Communication
PS	Start Input Update
SP	Stop Input Update

Table 27 Status transitions

Transitions	Operations
SO	Start Output Update
OS	Stop Output Update
OP	Stop Output Update/Stop Input Update
SI	Stop Input Update/Stop Mailbox Communication
OI	Stop Output Update/Stop Input Update/Stop Mailbox Communication

6.2 Configuration for operation in a controller

The services described in the previous section (e.g. PDO mapping etc.) are all operated by the controller (EtherCAT master). The communication-specific configuration of YukonDrive is performed on the basis of the supplied XML files by the master.

The configuration of control settings, scaling etc. can also be performed via the DriveManager. Alternatively all parameters can also be configured via the object directory.

7 Implemented CiA-402 functionality

The functions in this section relate to activation in the modes of operation of the CiA-402 profile

1 – Profile Position mode
3 – Profile Velocity mode
6 – Homing mode
7 – Interpolated Position mode
8 – Cyclic Synchronous Position mode (EtherCAT only)
9 – Cyclic Synchronous Velocity mode (EtherCAT only)
10 – Cyclic Synchronous Torque mode (EtherCAT only)

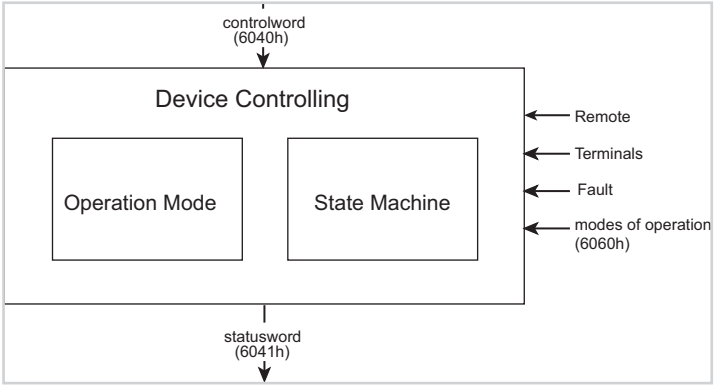
7.1 Device control and state machine

The drive is controlled via the DRIVECOM state machine defined in CiA-402 (see CiA-402 10.1.1 state machine). No remote signal is provided.

7.1.1 General information

The DEVICE CONTROL FUNCTION monitors all the functions of the controller. This function is subdivided into

- device control of the state machine
- operation mode function



Picture 13 Device controlling

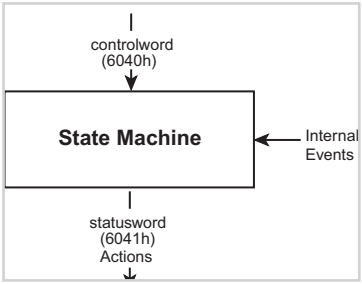
The status of the controller is controlled by way of the control word. The status of the controller is displayed in the STATUS WORD. In REMOTE MODE the controller is controlled directly from the CANopen network by PDO and SDO.

The state machine is controlled by the control word. The state machine is also influenced by internal events, such as errors.

7.1.2 State machine

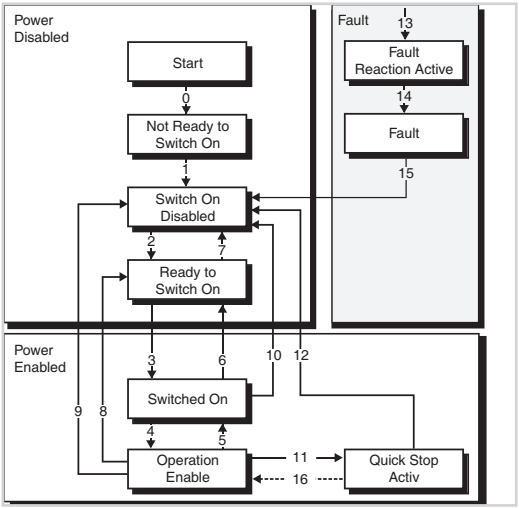
The state machine describes the CONTROLLER STATUS and the possible options for control by the master. A single status indicates a specific internal or external response. At the same time, the status of a controller restricts the possible control commands. For example, initiating a point-to-point positioning operation is only possible in the OPERATION ENABLE state.

States may change because of the control word or other internal events. The current status is displayed in the STATUS WORD. The state machine describes the controller status with regard to user commands and internal error messages.



Picture 14 State machine

7.1.3 Device states



Picture 15 State machine

The following device states are possible:

NOT READY TO SWITCH ON:

- Only low voltage is connected to the drive.
- The drive is initialised or is performing a self-test.
- If installed, the brake engages in this state.
- The drive function is deactivated.

SWITCH ON DISABLED: (Starting lockout)

- Drive initialisation is complete.
- Drive parameters have been set.
- Drive parameters have been changed.
- No power to device (for safety reasons).
- The drive function is deactivated.
- „STO (Safe Torque Off)" standstill and/or ENPO not active

READY TO SWITCH ON:

- Power is connected to the device.
- Drive parameters have been changed.
- Drive function is deactivated.

SWITCHED ON:

- Power is connected to the device.
- POWER AMPLIFIER is ready for operation.
- Drive parameters have been changed.
- The drive function is deactivated.

OPERATION ENABLE: (Technology ready)

- No errors were detected.
- Drive function is enabled and power is connected to motor.
- Drive parameters have been changed.
  (Refers to standard application of the drive.)

QUICK STOP ACTIVE:

- Drive parameters have been changed.
- QUICK STOP function being executed.
- Drive function is enabled and power is connected to motor.
- If the QUICK STOP OPTION CODE is set to 5 (remain in QUICK STOP ACTIVE status), you cannot quit the QUICK STOP ACTIVE status, but you can switch to OPERATION ENABLE status using the ENABLE OPERATION command.

FAULT REACTION ACTIVE:

- Drive parameters have been changed.
- An error has occurred in the device.
- The QUICK STOP function has been executed.
- The drive function is enabled and power is connected to motor.

FAULT:

- Drive parameters have been changed.
- An error has occurred and the fault reaction has been executed.
- Power disconnection and connection depends on the application.
- The drive function is deactivated.

Bit combinations of the DRIVECOM state machine

Device control commands:

The following bit combinations of control bits 0-3 and 7 form the device control commands for the status transitions of the state machine:

Table 28 Bit combinations

Command	Control word					Transitions
	7	3	2	1	0	
SHUTDOWN	0	X	1	1	0	2, 6, 8
POWER-UP	0	X	1	1	1	3
DISABLE POWER	0	X	X	0	1	7, 9, 10, 12
QUICK STOP	0	X	0	1	X	11
DISABLE OPERATION	0	0	1	1	1	5
ENABLE OPERATION	0	1	1	1	1	4
RESET FAULT	0 > 1	X	X	X	X	15

Device status table

Table 29 Bit combinations of the DRIVECOM state machine

Status	Status bit					
	6	5	3	2	1	0
NOT READY	0	X	0	0	0	0
STARTING LOCKOUT	1	X	0	0	0	0
READY	0	1	0	0	0	1
ON	0	1	0	0	1	1
OPERATION ENABLED	0	1	0	1	1	1
FAULT	0	X	1	0	0	0
FAULT REACTION ACTIVE	0	X	1	1	1	1
QUICK STOP ACTIVE	0	0	0	1	1	1

7.2 Option codes

The devices support option codes for four different options for shutting down the drive. These four options are:

- HALT function – interrupt an ongoing movement
- Controller disable function – stop movement by cancelling the controller enable (software)
- Quick-stop function – stop movement by initiating a quick stop
- Fault reaction function – stop movement in case of an error

For all variants, the option code sets the parameters for the desired device response.

Table 30 Option Codes

CANopen	Function	Supported settings
Object 605Ah	Quick stop option code	0 to 8
Object 605Bh	Shutdown option code	-1 to 1
Object 605Ch	Disable operation option code	0 and 1
Object 605Dh	Halt option code	0 to 4
Object 605Eh	Fault reaction option code	0 to 4

The objects form part of the data set as standard parameters of the devices.

NOTE: The quick-stop ramp is always executed with the smoothing preset for the driving profile ramps. The error stop ramp is always executed without smoothing, even when smoothing is programmed.

7.3 Device control objects

The following table lists the implemented objects for controlling the drive:

Table 31 Device control objects

Object no.	Object name	Object Code	Type	Attr.
0x6040	Control word	VAR	Unsigned16	rw
0x6041	Status word	VAR	Unsigned16	ro
0x605A	Quick_Stop_Option_Code 0: disable drive function 1: slow down on slow down ramp 2: slow down on quick stop ramp 3: slow down on the current limit 4: slow down on the voltage limit 5: slow down on slow down ramp and stay in QUICK STOP 6: slow down on quick stop ramp and stay in QUICK STOP 7: slow down on the current limit and stay in QUICK STOP 8: slow down on the voltage limit and stay in QUICK STOP	VAR	Integer16	rw
0x605B	Shutdown_Option_Code -1: Response as per Quick_Stop_Option_Code 0: disable drive function 1: slow down with slow down ramp; disable the drive function	VAR	Integer16	rw
0x605C	Disable_Operation_Option_Code 0: disable drive function 1: slow down with slow down ramp and then disable the drive function	VAR	Integer16	rw
0x605D	Halt_Option_Code 0: disable drive, motor is free to rotate 1: slow down on slow down ramp 2: slow down on quick stop ramp 3: slow down on the current limit 4: slow down on the voltage limit	VAR	Integer16	rw
0x605E	Fault_Reaction_Option_Code 0: disable drive, motor is free to rotate 1: slow down on slow down ramp 2: slow down on quick stop ramp 3: slow down on the current limit 4: slow down on the voltage limit	VAR	Integer16	rw
0x6060	Modes_Of_Operation 1: profile position mode 3: profile velocity mode 6: homing mode 7: interpolated position mode 8: cyclic sync position mode (EtherCAT ONLY) 9: cyclic sync velocity mode (EtherCAT ONLY) 10: cyclic sync torque mode (EtherCAT ONLY)	VAR	Integer8	wo
0x6061	Modes_Of_Operation_Display see 0x6060	VAR	Integer8	ro

7.4 Units and scalings, factor group

The DriveManager user interface offers a Scaling Wizard as a user-friendly means of configuring the scaling of mechanical and electrical units of variables necessary for control. The Wizard translates the application variables into representation of the parameters from the CiA-402 factor group. The parameters from the factor group are listed below, and can also be set directly by the user.

Correlations must be calculated externally and the final results entered in the relevant factor group parameter.

It is generally easier to have the Scaling Wizard calculate the parameter settings.

NOTE: The following objects are directly calculated in YukonDrive:
- Position factor
- Velocity encoder factor
- Acceleration factor

The calculation is based on the objects stored in the formulae (e.g. feed constant, gear ratio etc.). It is in fact possible to change these variables in DriveManager or via the bus, but they will be overwritten by the internal calculation as part of the control initialisation.

NOTE: In this section you will find an overview of the objects from the factor group and the underlying formulae for the calculation. You will find practical examples for the implementation of scaling in the Application Manual.

Factor group as per CiA-402:

Table 32 Factor Group

Object no.	Object name	Object Code	Type	Attr.
0x607E	Polarity	VAR	Unsigned8	rw
0x6089	Position_Notation_Index	VAR	Integer8	rw
0x608A	Position_Dimension_Index Only display for scaling block	VAR	Unsigned8	rw
0x608B	Velocity_Notation_Index	VAR	Integer8	rw
0x608C	Velocity_Dimension_Index Only display for scaling block	VAR	Unsigned8	rw
0x608D	Acceleration_Notation_Index	VAR	Integer8	rw
0x608E	Acceleration_Dimension_Index Only display for scaling block	VAR	Unsigned8	rw
0x608F	Position_Encoder_Resolution	VAR	Unsigned8	rw
0x6090	Velocity_Encoder_Resolution	ARRAY	Unsigned32	rw
0x6091	Gear_Ratio	ARRAY	Unsigned32	rw
0x6092	Feed_Constant	ARRAY	Unsigned32	rw
0x6093	Position_Factor	ARRAY	Unsigned32	rw
0x6094	Velocity_Encoder_Factor	ARRAY	Unsigned32	rw
0x6097	Acceleration_Factor	ARRAY	Unsigned32	rw

The factor group objects can be calculated and entered directly by the user, independently of the DriveManager Scaling Wizard. The corresponding encoder settings must be made however.

Calculation correlations for factor group parameters

Object 608Fh: Position encoder resolution
The position encoder resolution defines the relationship between the encoder and motor revolutions.

$$\text{position encoder resolution} = \frac{\text{encoder increments}}{\text{motor revolutions}}$$

Object 6090h: Velocity encoder resolution
The velocity encoder resolution defines the relationship between the encoder increments per second and motor revolutions per second

$$\text{velocity encoder resolution} = \frac{\text{encoder } \frac{\text{increments}}{\text{second}}}{\text{motor } \frac{\text{revolutions}}{\text{second}}}$$

Object 6091h: Gear ratio
Gear ratio defines the transmission ratio of a gear in relation to the motor.

It is defined as follows:

$$\text{gear ratio} = \frac{\text{motor shaft revolutions}}{\text{driving shaft revolutions}}$$

Object 6092h: Feed constant
The feed constant defines the feed per drive shaft revolution in position units.
This includes the gear if present.

$$\text{feed constant} = \frac{\text{feed}}{\text{driving shaft revolutions}}$$

Object 6093h: Position factor
The position factor converts the desired position (in position units) into the internal format (in increments).

$$\text{position factor} = \frac{\text{position encoder resolution} \bullet \text{gear ratio}}{\text{feed constant}}$$

Object 6094h: Velocity encoder factor
The velocity encoder factor converts the desired velocity (in velocity units) into the internal format (in increments).

$$\text{velocity encoder factor} = \frac{\text{velocity encoder resolution} \bullet \text{gear ratio} \bullet \text{position unit} \bullet \text{F velocity (notation index)}}{\text{feed constant} \bullet \text{velocity unit} \bullet \text{second} \bullet \text{F position (notation index)}}$$

An example of $F_{\text{velocity (notation index)}}$ or $F_{\text{position (notation index)}}$

would be 10^{2} or 10^{-6}

Object 6097h: Acceleration factor
The acceleration factor converts the acceleration (in acceleration units per second) into the internal format (in increments per second).

$$\text{acceleration factor} = \frac{\text{velocity unit} \bullet \text{velocity encoder factor}}{\text{acceleration unit} \bullet \text{second}}$$

Object 607Eh: Polarity
The position setpoint and position actual value are multiplied by 1 or -1 depending on the value of the polarity flag.
The same applies to the speed reference and actual speed value.
Please observe the operation of the object polarity as per CiA-402 V2.0.

Bits 0 to 5 = reserved (don't use)
Bit 6 = velocity polarity
Bit 7 = position polarity

NOTE: As in the case of the other objects in the factor group, changes in polarity only take effect if the control is switched off.

7.5 I/O map

The status of the drive controller's inputs and outputs can be determined using various objects. The following objects and parameters are implemented:

7.5.1 Object 60FDh – digital inputs

This object is implemented in compliance with device profile CiA-402. It allows digital input functions defined in the profile to be evaluated. That is, it does not offer an input map of existing physical inputs, but rather a function-related input map.

So the input to which, for example, a limit switch is connected is irrelevant. The bit that defines the state of the limit switch is permanently defined within the object.

Table 33 Object 60FDh – digital inputs

Bit	Assignment
0	Negative limit switch
1	Positive limit switch
2	Home switch
3 to 15	Reserved
16 to 31	Manufacturer-specific (curr. not implemented)
18	Status requirement for safe standstill
19	ENPO

7.5.2 Object 2079h – MPRO_INPUT_STATE

This manufacturer-specific object provides an input map of all the YukonDrive digital inputs. The object is mappable and transferable by PDO. The assignment is as follows:

Table 34 Object 2079h – MPRO_INPUT_STATE

Bit	Assignment
0	Status of input ENPO
1	Status of input ISD00
2	Status of input ISD01
3	Status of input ISD02
4	Status of input ISD03
5	Status of input ISD04
6	Status of input ISD05
7	Status of input ISDSH
8 to 15	Don't use
16	Status of input ISD06
17	Don't use
18	Status of input ISA00
19	Status of input ISA01
30 to 31	Don't use

7.5.3 Object 208Fh – MRPO_OUTPUT_STATE

This manufacturer-specific object provides an input map of all the YukonDrive digital outputs. The object is mappable and transferable by PDO. The assignment is as follows:

Table 35 Object 208Fh – MPRO_OUTPUT_STATE

Bit	Assignment
0	Status of output OSD00
1	Status of output OSD01
2	Status of output OSD02
3 to 5	Don't use
6	Status of output motor brake
7	Status of relay output

Table 36 Object 208Fh – MPRO_OUTPUT_STATE

Bit	Assignment
0	Status of output OSD00
1	Status of output OSD01
2	Status of output OSD02
3 to 5	Don't use
6	Status of output motor brake
7	Status of relay output
8 to 14	Don't use
15	Status of relay output "STO (Safe Torque Off)"

7.5.4 Setting digital outputs via field bus

In order to be able to set or reset digital outputs OSD00–OSD02 via the bus, the "MPRO_Output_FS_OSDxx" output selectors (parameter 122–124) must be configured for access via field bus. The two setting options provided for this are valid for all three digital outputs (OSD00, OSD01, OSD02) and are shown in the following table.

Table 37 Setting "MPRO_Output_FS_OSDxx" parameters (122–124

Setting	Description
(39) Output set via communication option in 1 ms cycle	Output set via communication option, updated in 1 ms cycle
(40) Output set via communication option in NC cycle	Output set via communication option, updated in control cycle (62.5 µs)

7.5.5 Object 60FE, digital outputs:

When the manufacturer-specific parameter “Function selector for digital output” is set to CAN (13), the associated output can be influenced by way of this object.

Bit assignment of the object	Bit
60FE assignment	
OSD00	16
OSD01	17
OSD02	18
OSD03	25
OSD04	26
OSD05	27

8 Operation modes CiA-402

8.1 CiA-402 compatible operation modes

Devices from the YukonDrive families support CiA-402 operation modes

- Profile position mode
- Profile velocity mode
- Homing mode
- Interpolated position mode
- Cyclic Synchronous Position mode (EtherCAT only)
- Cyclic Synchronous Velocity mode (EtherCAT only)
- Cyclic Synchronous Torque mode (EtherCAT only)

The operation mode is switched via CANopen object 6060h modes of operation. This switch is possible in “Operation enable” (power to motor) status. The current operation mode is indicated in the CANopen object 6061h modes of operation display.

8.1.1 Configuring YukonDrive for activation via CiA-402

For activation via CANopen (or CoE – EtherCAT) as per CiA-402 profile, the following parameters must be set in the device:

Table 38 Configuring YukonDrive

No.	Name	Function	Setting
159	MPRO_CTRL_SEL	Control location selector	CiA-402
165	PRO_REF_SEL	Setpoint selector	CiA-402

These parameters can be found under “Motion Profile” --> “Basic Settings”

If the drive is operated in an operation mode in which the internal profile generator is inactive and cyclic setpoints are transferred (e.g. cyclic synchronous position mode), the interpolation time must be configured.

Table 39 Configuring YukonDrive

No.	Name	Function
306	CON_IpRefTs	Cycle time of setpoints in IP mode

The interpolation time CON_IpRefTs represents the cycle time in which setpoints from a higher-level controller are expected.

8.1.2 Control word CiA-402

Object 6040h-control word

The object is also mapped in the parameter P 2208-MP_Controlword. The control word contains bits for:

- status control
- control of operating modes and
- manufacturer-specific options.

The bits in the control word are defined as follows:

Table 39 Control word CiA-402

15 11	10 9	8	7	6 4	3	2	1	0
Manufacturer-specific	reserved	Stop	Fault Reset	Operation mode-specific	Enable operation	Quick stop	Enable voltage	Switch on
O	O	O	M	O	M	M	M	M

MSB LSB

O – Optional M – Mandatory

Bits 0–3 and 7:

DEVICE CONTROL COMMANDS are triggered by the following schema in the control word:

Table 40 Device control commands

Command	Bit of the control word					Transitions
	Fault reset	Enable operation	Quick stop	Enable voltage	Switch on	
Shutdown	0	X	1	1	0	2, 6, 8
Switch on	0	0	1	1	1	3*
Switch on	0	1	1	1	1	3**
Disable voltage	0	X	X	0	X	7, 9, 10, 12
Quick stop	0	X	0	1	X	7, 10, 11
Disable operation	0	0	1	1	1	5
Enable operation	0	1	1	1	1	4, 16
Fault reset		X	X	X	X	15

Bits marked X are irrelevant.

* ... In SWITCHED ON status the drive executes the functionality of this state.

** .. There is no functionality in SWITCHED ON status. The drive does not do anything in this state.

Bits 4–6 and 8

Bits 4–6 and 8 are interpreted differently according to the active operation mode (“modes of operation display” object).

Table 41 Mode-specific bits in the control word

Operation mode							
Bit	Profile position mode	Profile velocity mode	Homing mode	Interpolated position mode	Cyclic synchronous position mode (Ether-CAT)	Cyclic synchronous velocity mode (Ether-CAT)	Cyclic synchronous torque mode (Ether-CAT)
4	New setpoint	reserved	Homing operation start	Enable IP mode-	reserved	reserved	reserved
5	Change set immediately	reserved	reserved	reserved	reserved	reserved	reserved
6	abs/rel	reserved	reserved	reserved	reserved	reserved	reserved
8	Stop	Stop	Stop	Stop	reserved	reserved	reserved

Use of the specific bits is explained in more detail in the sections on the operation modes.

Bits 7 and 11–15

Bit	Name	Value	Description
7	Fault Reset	0 → 1	Fault reset
11			No function
. . .			No function
15			No function

8.1.3 Status word CiA-402

Object 6041h status word

The content of the object is also mapped in parameter P 2209 – MP_Statusword. The status word indicates the current status of the drive. It contains the following bits for:

- current state of the device,
- status of the operation mode and
- manufacturer-specific options.

Status word bits

Table 42 Bits in the status word

Bit	Description	M/O
0	Ready to switch on	M
1	Switched on	M
2	Operation enabled	M
3	Fault	M
4	Voltage enabled	M
5	Quick stop	M
6	Switch on disabled	M
7	Warning	O
8	Manufacturer-specific	O
9	Remote	M
10	Target reached	M
11	Internal limit active	M
12 – 13	Operation mode-specific	O
14 – 15	Manufacturer-specific	O

Bits 0–3, 5 and 6:

These bits indicate the STATUS of the controller.

Table 43 Device state bits in the status word

Value (binary)	State
xxxx xxxx x0xx 0000	Not ready to switch on
Xxxx xxxx x1xx 0000	Switch on disabled
Xxxx xxxx x01x 0001	Ready to switch on
Xxxx xxxx x01x 0011	Switched on
Xxxx xxxx x01x 0111	Operation enabled
Xxxx xxxx x00x 0111	Quick stop active
Xxxx xxxx x0xx 1111	Fault reaction active
Xxxx xxxx x0xx 1000	Fault

Bit 4: Voltage enabled

Power supply is connected.

Bit 5: Quick stop

In the LOW state this bit indicates that the controller is executing a “quick stop”. Bits 0, 1 and 2 of the status word are set to 1 when the drive is ready for operation. The other bits indicate additional states of the drive, such as execution of a “quick stop”.
In the event of an error the FAULT bit is set.

Bit 7: Warning

Warnings, such as temperature limits, are indicated in bit 7. The device state does not change when warnings are issued. For more information on the warning given, refer to the FAULT CODE.

Bit 8: Manufacturer-specific

Currently not used.

Bit 9: Remote

Currently not used.

Bit 10: Target reached

The bit is automatically set when a SETPOINT is reached. The setpoint depends on the OPERATING MODE. A change to the setpoint by the master changes this bit. With “quick stop” OPTION CODE 5, 6, 7 or 8, this bit is set when the “quick stop” ends. This bit is also set at a standstill in response to a STOP request.

Bit 11: Internal limit active

This bit is set when internal limits are reached. This bit is dependent on OPERATION MODE.

Bits 12 and 13:

These bits are dependent on OPERATION MODE – see following section.

The following table provides an overview:

Table 44 Mode-specific bits in the control word

Operation mode							
Bit	Profile position mode	Profile velocity mode	Homing mode	Interpolated position mode	Cyclic synchronous position mode (Ether-CAT)	Cyclic synchronous velocity mode (Ether-CAT)	Cyclic synchronous torque mode (Ether-CAT)
12	Setpoint acknowledge	Speed	Homing attained	IP mode active	Target position ignored	Target velocity ignored	Target torque ignored
13	Following error	Max. slippage error	Homing error	reserved	Following error	reserved	reserved

Bits 14 and 15:

These bits are implemented specific to the manufacturer; explanatory notes for them are given in the sections on the various operation modes.

8.2 Operation modes with profile generation in drive

For operation modes with profile generation in the drive, the controller merely transmits a target position or speed for the movement to the drive controller. How the drive controller reaches this position/speed – i.e. the configuration of the driving profile (e.g. trapezoidal, triangular/steepness of ramps etc.) – is determined and executed entirely by the drive controller.

8.2.1 Profile velocity mode

This operation mode (mode of operation = 3) is used to activate the device at a velocity setpoint as per the CiA-402 profile. The drive is in speed control in this operation mode.

The units, setpoint and ramp variables are derived from the factor group settings. See also section 7.4 “Units and scalings”.

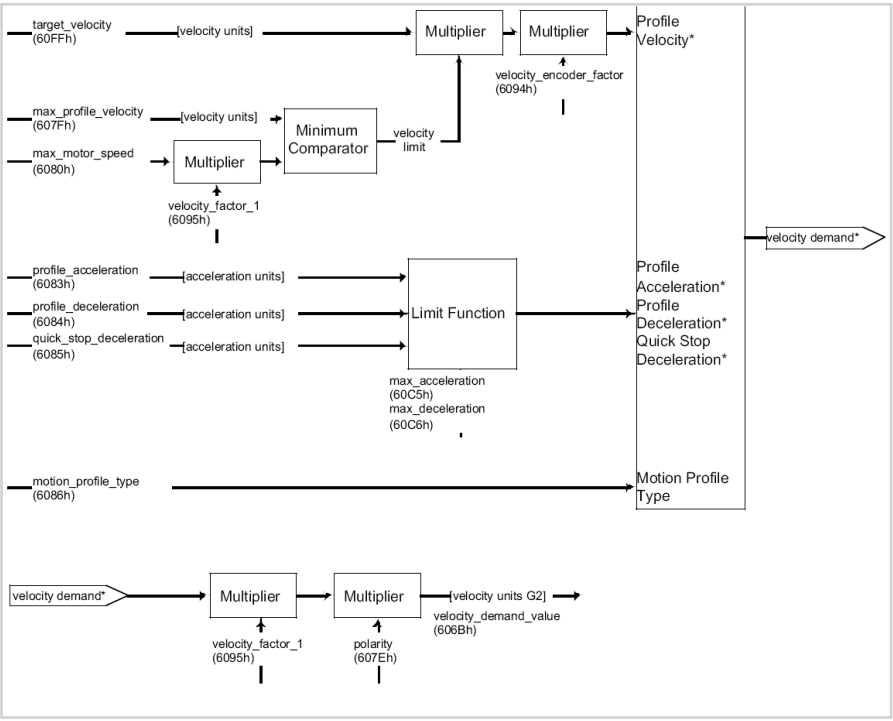
The device supports the following objects for this operation mode:

Table 45 Profile velocity mode

Object no.	Object name	Object code	Type
0x606C	Velocity actual value	VAR	Int32
0x60FF	Target velocity	VAR	Int32
0x6094	Velocity encoder factor	ARRAY	Int32
0x6083	Profile acceleration	VAR	Int32
0x6084	Profile deceleration	VAR	Int32
0x6085	Quick stop deceleration	VAR	UInt32
0x607E	Polarity	VAR	UInt8

Note: In addition to the objects listed in the table, object 0x6064 “Position Actual Value” is also updated cyclically in profile velocity mode.

Structure of operation mode



Picture 16 Structure of profile velocity mode

Mode-dependent bits in the control word
The structure presented below is based on this operation mode:

Table 46 Profile velocity mode bits in the status word

Object no.	Object name	Object code	Type
8	Stop	0	Execute the motion
		1	Stop axle

8.2.2 Homing mode

This operation mode (mode of operation = 6) is used for homing a position-controlled axle. The drive executes a movement according to the programmed homing method.

Note: The Touch probe function enables control-led homing of the drive. See section 10.1.

The various homing methods differ in the integration of the hardware limit switch, home switch and index signal into the encoder system. It should be noted here that appropriate digital inputs should be configured for limit switch and home switch functionality:

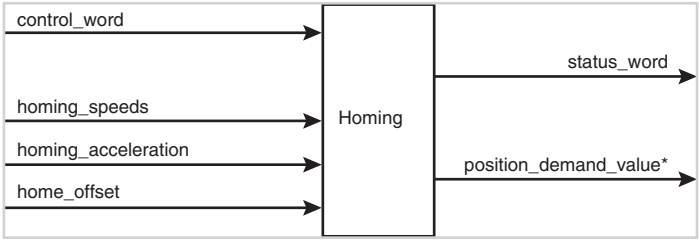
- Limit switch function
- LCW – right-hand hardware limit switch
- LCCW – left-hand hardware limit switch
- HOMSW – home switch

The following objects are supported by the device for this operation mode:

Table 47 Homing Mode

Object no.	Object name	Object code	Type	Attr.
0x607C	Home_Offset	VAR	Integer32	rw
0x6098	Homing_Method	VAR	Integer8	rw
0x6099	Homing_Speeds *	ARRAY	Unsigned32	rw
0x609A	Homing_Acceleration	VAR	Unsigned32	rw

* 0x6099.01 – quick jog
0x6099.02 – slow jog



Picture 17 Homing function

YukonDrive supports all 35 homing methods defined in CiA-402.
The individual homing methods' functions and movement sequences are described in the device application manuals.

Home offset:

The HOME OFFSET object is the difference between position 0 of the application and the HOME POSITION found during homing. It is represented in position units. At the end of a homing run the HOME OFFSET is added to the HOME POSITION found. All subsequent absolute positioning operations relate to this new home position.

The homing method and the associated properties can be changed in two ways.
Homing can be changed either via DriveManager or via CAN.

For configuration via CANopen the objects of the homing mode can be directly addressed. For example, for a change to the homing method, object 0x6098 can be changed.

Mode-specific bits in the control word

Bit 4 – HOMING OPERATION START

Bit 8 – STOP

Table 48 Homing Mode bits in the control word

Bit	Name	Value	Description
4	Homing operation start	0	Homing mode inactive
		0→ 1	Start homing mode
		0	Homing mode active
8	Stop	1 → 0	Interrupt homing mode
		0	Execute the instructions of bit 4
		1	Stop axle with profile deceleration

Mode-specific bits in the status word

Bit 10 – TARGET REACHED
Bit 12 – HOMING ATTAINED
Bit 13 – HOMING ERROR
Bit 14 – ROT_0

Table 49 Homing mode bits in the status word

Bit	Name	Value	Description
10	Target reached	0	Stop = 0: Home position not reached Stop = 1: Axle decelerates
		1	Stop = 0: Home position reached Stop = 1: Axle has velocity 0
12	Homing attained	0	Homing mode not yet completed
		1	Homing mode carried out successfully
13	Homing error	0	No homing error
		1	Homing error occurred; Homing mode not carried out successfully The error cause is found by reading the error code
14	ROT_0	1	Axle at standstill Speed is much lower than parameter 745 MON_REFWINDOW

8.2.3 Profile position mode

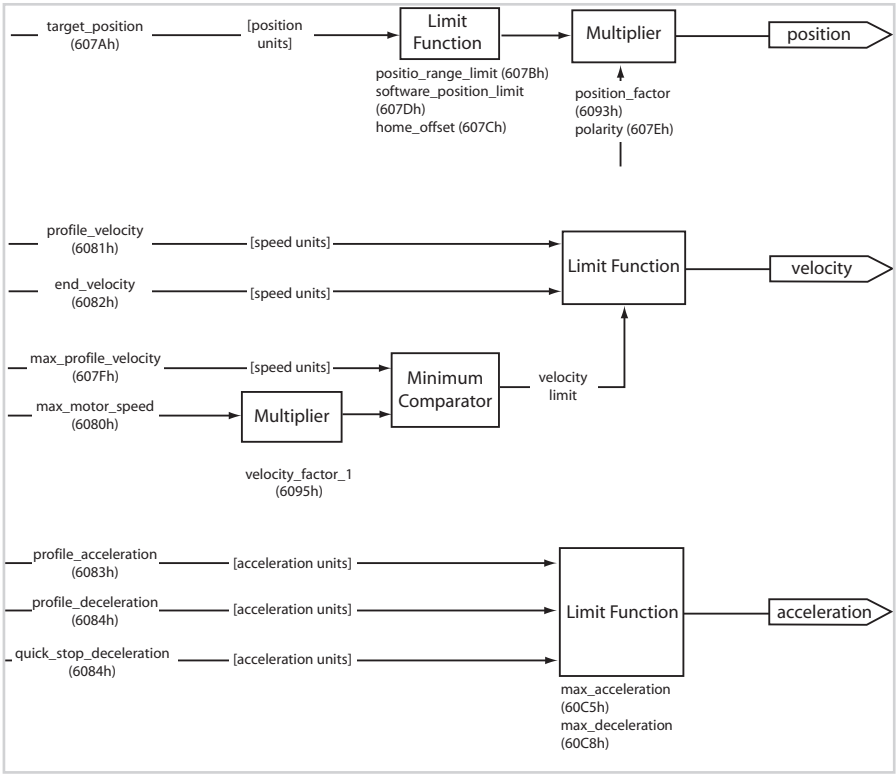
In this operation mode (mode of operation = 1) the axle executes relative or absolute single positioning movements.

Table 50 Profile position mode

Object no.	Object name	Object code	Type	Attr.
0x607A	Target_Position	VAR	Integer32	rw
0x607d	Software position limit	ARRAY	Integer32	rw
0x6081	Profile_Velocity	VAR	Unsigned32	rw
0x6083	Profile_Acceleration	VAR	Unsigned32	rw
0x6084	Profile_Deceleration	VAR	Unsigned32	rw
0x6085	Quick stop deceleration	VAR	Unsigned32	rw
0x6064	Position actual value	VAR	Integer32	r
0x607E	Polarity	VAR	Unsigned8	rw

Units of the parameters are set by way of the Scaling Wizard or the objects from the factor group.

Structure of operation mode



Picture 18 Structure of profile position mode

Mode-specific bits in the control word

Bit 4 – New setpoint
Bit 5 – Change set immediately
Bit 6 – abs/rel
Bit 8 – Stop

Table 51 Profile position mode bits in the control word

Bit	Name	Value	Description
4	New setpoint	0	Does not assume target position
		1	Assume target position
5	Change set immediately	0	Finish the current positioning and then start the next positioning
		1	Interrupt the actual positioning and start the next positioning
6	abs/rel	0	Target position is an absolute value
		1	Target position is a relative value
8	Stop	0	Execute positioning
		1	Stop axle with profile deceleration (if not supported with profile acceleration)

Mode-specific bits in the status word

Bit 10 – Target reached
Bit 12 – Setpoint acknowledge
Bit 13 – Following error
Bit 14 – ROT_

Table 52 Profile position mode bits in the status word

Bit	Name	Value	Description
10	Target reached	0	Stop = 0: Target position not reached Stop = 1: Axle decelerates
		1	Stop = 0: Target position reached Stop = 1: Velocity of axle is 0
12	Setpoint acknowledge	0	Trajectory generator has not assumed the positioning values (yet)
		1	Trajectory generator has assumed the positioning values

Table 53 Profile position mode bits in the status word

Bit	Name	Value	Description
13	Following error	0	No following error
		1	Following error
14	ROT_0	1	Axle at standstill speed is much lower than parameter 745 MON_REFWINDOW

Functional description

This OPERATION MODE supports two different options for target position input.

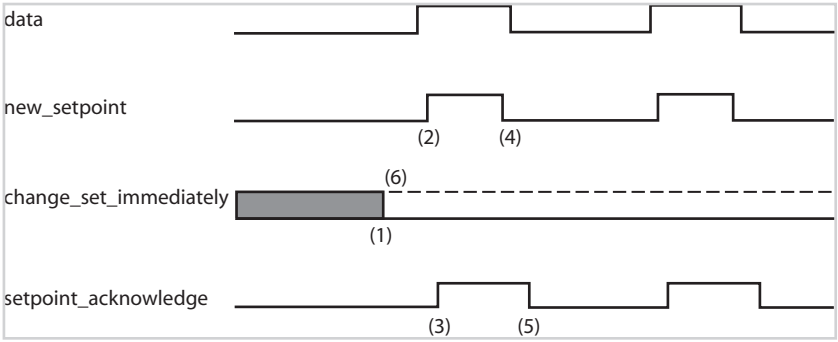
SET OF SETPOINTS:

When the target position is reached, the drive directly approaches the next target position; the axle is not stopped when the first target position is reached.

SINGLE SETPOINT:

When the target position is reached the drive indicates the fact to the master. Then the drive receives a new setpoint. At each target position the drive is stopped before being moved on to the next target position.

The two options are controlled via the timing of the NEW SETPOINT and CHANGE SET IMMEDIATELY bits in the control word and the SETPOINT
ACKNOWLEDGE bit in the status word. These bits allow a new positioning
operation to be initiated even while the current one is ongoing.



Picture 19 Setpoint transmission from a host computer

If the 'CHANGE SET IMMEDIATELY' bit is set to "0" (solid line in above diagram) a SINGLE SETPOINT is expected by the drive (1).

When the setpoint has been transmitted to the drive, the master activates the positioning by setting the "new setpoint" bit in the control word (2). The drive responds by setting the "setpoint acknowledge" bit in the status word (3) once the new data has been detected and saved. Now the master can delete the "new setpoint" bit (4). Then the drive deletes the "set-point acknowledge" bit to signal that a new setpoint is accepted (5). In the diagram the mechanism initiates a speed 0 on reaching the target position at time t1. After the message indicating the target position has been reached, the next target position can be initiated at time t2.

8.2.4 Velocity mode (V/F mode)

This operation mode (mode of operation = 2) is used to control the drive in frequency-regulated mode (V/F mode).

The units, setpoint and ramp variables are derived from the factor group settings. See also section 5.4 "Units and scalings".

Yukon Drive supports the following objects in this operation mode:

Table 54 Velocity mode

Object no.	Object name	Object code	Type
0x6042	vl target velocity	VAR	Integer16

Table 55 Velocity mode

Object no.	Object name	Object code	Type
0x6046	vl min./max. amount	ARRAY	Unsigned32
0x6048	vl velocity acceleration	ARRAY	Unsigned32
0x6049	vl velocity deceleration	ARRAY	Unsigned32

In this operation mode, the device must be scaled in the unit Hertz (Hz). This requires the following settings using the Scaling Wizard in the standard/CiA-402 area:

Position → rev

Speed → rev/s (=> 1/s => Hz)

Acceleration → rev/s/s

The limits then also have to be set. Specifically, these are:

Table 55 Limits in V/F mode

Object no.	Object name	Description	
0x6046	vl min./max. amount	Index	
		0	Min. speed in user unit
		1	Max. speed in user unit
0x6048	vl velocity acceleration	Index	
		0	Speed change in user unit
		1	Per time unit
0x6049	vl velocity deceleration	Index	
		0	Speed change in user unit
		1	Per time unit

8.3 Cyclical operation modes, profile generation in the controller

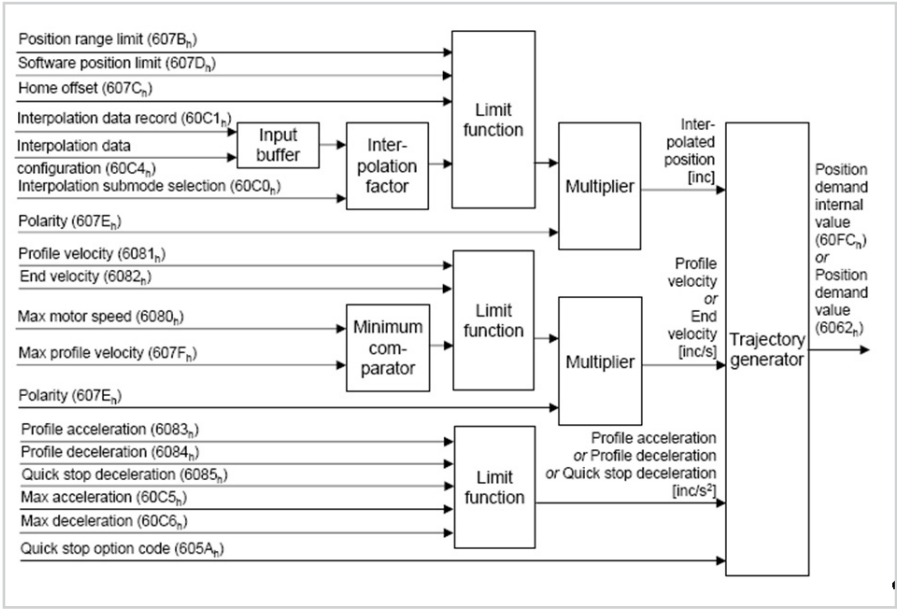
In the cyclical operation modes described below, the profile generation takes place in the controller; the drive's internal profile generator is not active. The drive interpolates between the controller setpoints transmitted cyclically (according to position, speed, torque operation mode).

8.3.1 Interpolated position mode

The "interpolated position mode" operation mode (mode of operation = 7) is a further option, alongside profile position mode, for positioning axles via CANopen. It is used for co-ordinated movement of multiple axles (or a single axle) via one control.
In Interpolated Position mode, though, the driving profile is created entirely by the control. It cyclically transmits roughly interpolated position values between which the drive controller handles the fine interpolation (e.g. linear). Accordingly,
the profile for the axle to follow is determined through the change in target positions for each time unit. Position control is therefore implemented not only in the drive, but also at the control level.
The following objects are supported by the device for this operation mode:

Table 56 Supported Objects

Object no.	Object name	Object code	Type
0x60C0	Interpolation sub mode select	VAR	Integer16
0x60C1	Interpolation data record	ARRAY	Integer32
0x60C2	Interpolation time period	RECORD	Index0: Unsigned8 Index1: Integer8



Picture 20 Structure of interpolated position mode

Mode-specific bits in the control word

Table 57 Mode specific bits in the control word

Bit	Name	Value	Description
4	Enable IP mode	0	Interpolated position mode inactive
		1	Interpolated position mode active
8	Stop	0	Execute the instruction of bit 4
		1	Stop axle

Mode-specific bits in the status word

Table 58 Mode specific bits in the status word

Bit	Name	Value	Description
10	Target reached	0	Stop = 0: Position not (yet) reached Stop = 1: Axle decelerates
		1	Stop = 0: Position reached Stop = 1: Axle has velocity 0
12	IP mode active	0	Interpolated position mode inactive
		1	Interpolated position mode active
14	Axle synchronised	0	Axle not synchronised
		1	Axle synchronised

8.3.2 Cyclic Synchronous Position mode (EtherCAT only)

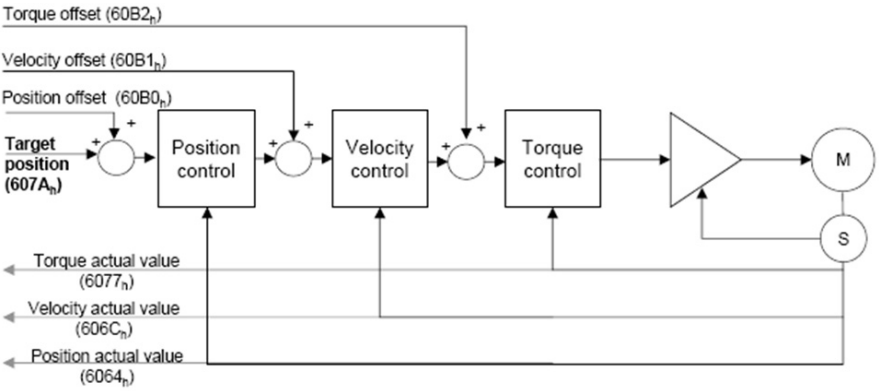
In this operation mode (mode of operation = 8) the controller cyclically provides the position setpoints for the drive. The position, speed and current are controlled by the drive.

As an option, an additional speed and torque setpoint can be transmitted as a pre-control value.

The following objects are supported by the device for this operation mode:

Table 59 Supported objects

Object no.	Object name	Object code	Type
0x607A	Target position	VAR	Integer32
0x60B1	Velocity offset	VAR	Integer32
0x60B2	Torque offset	VAR	Integer16



Picture 21 Overview of cyclic synchronous position mode

Mode-specific bits in the status word

Table 60 Mode-specific bits in the status word

Bit	Name	Value	Description
12	Target position	0	Target position ignored
		1	Target position shall be used as input
13	Following error	0	No following error
		1	Following error

8.3.3 Cyclic Synchronous Velocity mode (EtherCAT only)

In this operation mode (mode of operation = 9) the controller cyclically transmits speed setpoints to the drive, which controls the speed and current. As an option, an additional speed setpoint and additional torque setpoint for torque pre-control can be transmitted by the controller.

The following objects are supported by the device for this operation mode:

Table 61 Supported objects

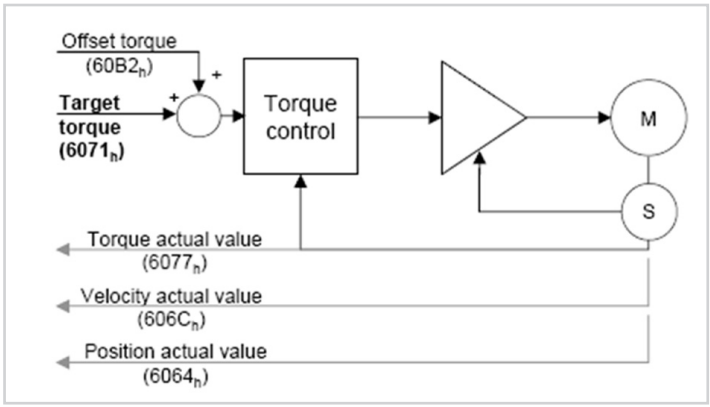
Object no.	Object name	Object code	Type
0x60FF	Target velocity	VAR	Integer32
0x60B1	Velocity offset	VAR	Integer32
0x60B2	Torque offset	VAR	Integer16

8.3.4 Cyclic Synchronous Torque mode (EtherCAT only)

In this operation mode (mode of operation = 10) the controller cyclically transmits torque setpoints to the drive, which controls the current. As an option, an additional torque setpoint can be transmitted.

Table 62 Supported objects

Object no.	Object name	Object code	Type
0x6071	Target torque	VAR	Integer16
0x60B2	Torque offset	VAR	Integer16



Picture 22 Overview of cyclic synchronous torque mode
Mode-specific bits in the status word

Table 63 Mode-Specific bits in the status word

Bit	Name	Value	Description
12	Target torque	0	Target torque ignored
		1	Target torque shall be used as input

8.3.5 External pre-control of speed/torque

When using the drive controller in the cyclic synchronous position mode (CSP, see section 9.3.2) or cyclic synchronous velocity mode (CSV, see section 9.3.3), it is possible to specify external pre-control values for the speed and torque via the controller. The internal pre-control function in the drive must be deactivated for this. The relevant settings can be found in the tables below:

Table 64 CiA402 objects for external pre-control

Object no.	Object name	Data type	Scaling
0x60B1	Velocity offset	Integer32	As per scaling for speeds (CiA402 factor group)
0x60B2	Torque offset	Integer16	In [‰] based on rated motor torque in object 0x6076, i.e. a value of 1000 corresponds to the rated motor torque.

Table 65 YukonDrive device parameter

Parameter	Function	Value
375 – CON_IP_SFF-Scale	Scaling of speed pre-control	0–100% based on the pre-control value
376 – CON_IP_TFFScale	Scaling of torque pre-control	0–100% based on the pre-control value
379 – CON_IP_FFMode	Switchover of pre-control sources and specific setpoint formats	See individual subindexes
• Subindex 0	PositionHighResolution	0 = 32-bit position setpoint (default)
• Subindex 1	Source for speed pre-control values	0 = internal pre-control (default) 1 = external pre-control
• Subindex 2	Source for torque pre-control values	0 = internal pre-control (default) 1 = external pre-control

Types of interpolation:

When using external pre-control via EtherCAT, both linear and cubic or spline interpolation can be applied. The type of interpolation is set via parameter P 370 – CON_IP. However, do NOT use the setting "SplineExtFF". This type of interpolation is reserved for a different operation mode.

Checking pre-control variables in DriveManager 5

You can check the transmitted external pre-control variables in YukonDrive in 2 ways:

- The objects for pre-control can be found in the CANopen/EtherCAT subject area as device parameters
- The variables nref_Ext (external speed pre-control) and mref_Ext (external torque pre-control) can be recorded with the internal oscilloscope.

Table 66 Emergency telegram

Byte	0	1	2	3	4	5	6	7
Bit:	0 ... 7	8 ... 15	16 ... 23	24 ... 39		40 ... 47	48 ... 63	
Profile	Device profile CiA-402			Drive controller				
Error	Emergency error code as per CiA-402		Error register (object 1001 h)	Error number	Error location	Operating hours meter (in full hours)		

The decisive factors for rapid localisation are the error code and error location. Byte 3 of the emergency telegram contains the error code, which provides an initial categorisation of the cause of the error. The precise cause of the error is specified by the error location in byte 4. Bytes 5, 6 and 7 contain the internal operating hours meter of the device.

CANopen errors – i.e. incorrect configurations, bus disturbances etc. – are indicated by error code 0xFF00.
Note: When an error occurs the controller executes a response as per the parameterised error response. These can be set separately for individual errors.

Note: The status indicators of the 7-segment display are explained in the application manual.

Note: A full list of all error messages from YukonDrive, including assignment of the corresponding emergency code, can be found in the YukonDrive application manual.

9.1 Error acknowledgement, general

Device errors can be acknowledged by the following mechanisms:

- Control word bit 7, edge-controlled
- Control input with programmed reset functionality
- Hardware enable ENPO to control terminal
- Operation via two buttons
- Drive Manager user interface
- Writing value 1 to parameter 153 MPRO_DRVCOM_FaultReset via the control unit or bus system

NOTE: For a detailed list of all error messages and remedial measures, please refer to the YukonDrive application manual on our product CD.

9.2 Error acknowledgement via bus system

Another option is available via the object 6040 h control word:

Draft 402	6040h	VAR	Control word	Integer16	rw	M

An error acknowledgement is executed by a rising edge at bit 7 in the control word. Resetting of the error is signalled by transmission of the following emergency message:

Table 67 Error acknowledgement

ID	Data bytes	Description
Emergency	00 00 00 00 00 00 00 00	Emergency message acknowledgement error

If the cause of the error is not eliminated, the drive controller returns to error status after transmission of another emergency message.

10 Technology functions

10.1 Touch probe

Positions of the drive can be recorded on the basis of certain input signals using the Touch probe function.
Possible input signals are:

- Digital input ISD05
- Digital input ISD06
- Index signal

It is possible to switch between different implementations using parameter 2285 "Touch probe function selector".

- CiA 402 implementation (not yet implemented)
- Manufacturer-specific implementation

10.1.1 Description of manufacturer-specific implementation

In order to be able to use this function, parameter P 2285 Touch probe function selector must first be set to 2 = "BECK2" (the setting "BECK1" is currently not supported). If signals are to be recorded via the two digital inputs ISD05 and ISD06, these will have to be configured using parameters P 106 + P 107 MPRO_Input_FS_ISD0x as measuring buttons (setting 15). These parameters can be found in the subject area "Configuration of inputs/ outputs → Digital inputs".

Finally, the following objects also have to be mapped:

RxPDO	0x60B8 Touch probe function
TxPDO	0x60B9 Touch probe status 0x60BA Touch probe pos1 pos value

Object 0x60B8 "Touch probe function" is used to specify whether the touch probe function is to be triggered on the falling edge, rising edge or on both edges of the respective signal. Setting the corresponding bit (0–4) activates the respective function (edge-controlled).

The reading of the stored position is controlled by bits 8 – 12. On receipt of the configured signal, a new measurement must be started by resetting and then reconfiguring the corresponding bit.

Table 68 Object 0x60B8: Touch probe function

Bit	Value (bin)	Value (hex)	Description
0	00000000 00000001	xx01	Enable external latch 1 (positive rise) via Touch probe module
1	00000000 00000010	xx02	Enable external latch 1 (negative rise) via Touch probe module
2	00000000 00000100	xx04	Enable external latch 2 (positive rise)
3	00000000 00001000	xx08	Enable external latch 2 (negative rise)
4	00000000 00010000	xx10	Enable internal latch C (positive rise) via MC_Home module
5–7	-	-	reserved
8–12	00000001 00000000	01xx	Read external latch 1 (positive rise) via Touch probe module
	00000010 00000000	02xx	Read external latch 1 (negative rise) via Touch probe module
	00000011 00000000	03xx	Read external latch 2 (positive rise)
	00000100 00000000	04xx	Read external latch 2 (negative rise)
	00000101 00000000	05xx	Read internal latch C (positive rise) via MC_Home module
13–15	-	-	reserved

Object 0x60B9 returns the status of the Touch probe function. If a signal activated by object 0x60B8 has been registered, this is indicated in the status word by setting the corresponding bit (0–4).

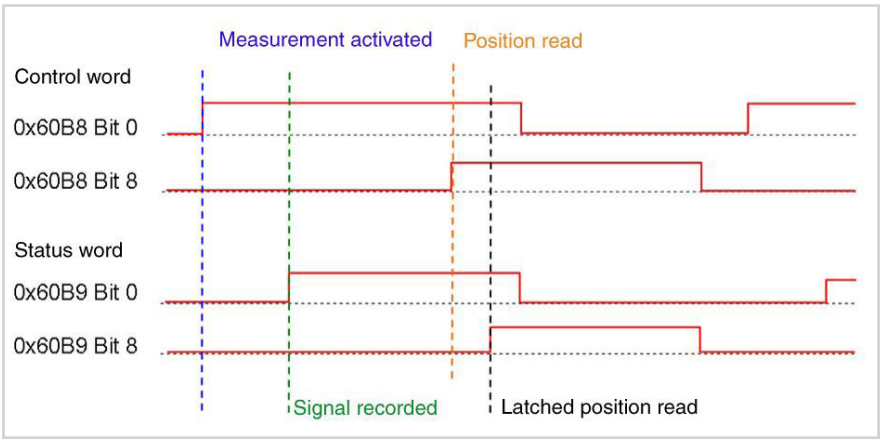
Table 69 Object 0x60B9: Touch probe status

Bit	Value (bin)	Value (hex)	Description
0	00000000 00000001	xx01	External latch 1 valid (positive rise) via Touch probe module
1	00000000 00000010	xx02	External latch 1 valid (negative rise) via Touch probe module
2	00000000 00000100	xx04	External latch 2 valid
3	00000000 00001000	xx08	External latch 2 valid
4	00000000 00010000	xx10	Internal latch C valid (positive rise) via MC_Home module
5–7	-	-	reserved
8–11	00000001 00000000	01xx	Acknowledge value external latch 1 (positive rise) via Touch probe module
	00000010 00000000	02xx	Acknowledge value external latch 1 (negative rise) via Touch probe module
	00000011 00000000	03xx	Acknowledge value external latch 2 (positive rise)
	00000100 00000000	04xx	Acknowledge value external latch 2 (negative rise)
	00000101 00000000	05xx	Acknowledge value internal latch C (positive rise) via MC_Home module
12–15	00010000 00000000	1xxx	reserved
	00100000 00000000	2xxx	reserved
	01000000 00000000	4xxx	reserved
	10000000 00000000	8xxx	reserved

The stored position is written to object 0x60BA after bits 8–11 have been set in the status word.

Time flowchart:

The time sequence of a measurement is shown using the example of measuring button ISD05 and the corresponding bits. The time sequence applies analogously to the other configurable signals.



Picture 23 Time sequence of Touch probe function

10.1.2 Controlled homing

The Touch probe function enables control-led homing of an axis. All the input signals described in section 10.1.1 can be used for this.

10.2 Indexing table function

The indexing table function is set in DriveManager 5 using the Scaling Wizard in the Movement Profile ⍰ Scalings/Units subject area.

A detailed description of the function can be found in the YukonDrive application manual.

11 EDS file, object directory, parameter list

11.1 EDS file, object directory

An EDS file is available for integrating the devices into the CAN master. The file is shipped with the firmware. It contains all the CAN objects of the drive controllers.

NOTE: YukonDrive has parameters with default values in the device that may deviate from the default values in the EDS file. These are power stage-specific parameters with contents that are dependent on the size.
Examples of such parameters are:
Para P 0302 – CON_SwitchFreq
Para P 0307 – CON_VoltageSupply
Para P 0651 – DV_CAL_VDC

Picture 24 Appendix: Glossary

CAL:	(CAN Application Layer). CiA protocol, primarily describes the way in which variables are transmitted without defining their function or content. Subsets: CMC: (CAN based Message Specification). Sets out the definition described above. Is accepted by most CAN suppliers. LUST conforms to this definition. NMT: (Network Management). Required for masters in the CAN system. Not implemented because drive controllers are always slaves and have no "control function". LMT: (Layer Management). See NMT DBT: (Identifier Distributor). See NMT
CANopen:	Based on CAL definition Corresponds to CiA Draft Standard 301 Expands the CAL definition to include function and unit assignment of the predefined variables This definition is being drafted by CiA and various user groups (MOTION for drive technology and I/O for inputs/outputs) (e.g. variable for torque in Nm).
CiA:	(CAN in Automation). CAN bus user group, generally defines a protocol for automation.
General points on the various protocol definitions	
CAL:	Mainly in use in Europe, Yukon Drive has currently implemented a protocol which can be activated by a CAL master. Initialisation has been simplified compared to CAL (CCDA), e.g. addressing via jumper, although this does not affect operation.
DeviceNet:	Mainly in the USA (corresponds to CAL definition).

Harmonic Drive AG

Harmonic Drive AG
Hoenbergstraße 14
65555 Limburg/Lahn
P.O. Box 1652
Germany
☎ + 49 (0) 6431 5008-0
💻 + 49 (0) 6431 5008-119
info@harmonicdrive.de
www.harmonicdrive.de

Änderungen vorbehalten.
We reserve the right to make technical changes without prior notice.

1003376 09/2011